On Interactive Activation and Theoretical Ecology

Anonymous authors

March 25, 2025

Abstract

Interactive Activation (IA) and generalized Lotka-Volterra (gLV) are two successful models in cognitive science and theoretical ecology, respectively. We have recently merged the two models into a new one, gLIA, that is more tractable analytically and can capture some of the basic cognitive phenomena covered by IA. Here we show that gLIA can be trained efficiently, with stable dynamics observed empirically, and proved Lyapunov stability when the weight matrix is assumed symmetric. We argue that these results warrant further investigation into gLIA as a dynamical model of cognition.

Keywords: Interactive Activation; Lotka-Volterra; Dynamical systems; Cognition; Theoretical Ecology.

1 Introduction

In cognitive modeling, few frameworks have enjoyed as much success and longevity as Interactive Activation (McClelland & Rumelhart, 1981). IA's key strength is to confront head-on the daunting dynamical nature of cognitive processes, by a couple of equations that govern neural network interactions at the millisecond timescale. Not only has this approach proved fertile in different fields of cognition (McClelland, Mirman, Bolger, & Khaitan, 2014), but its explanatory power can also be very deep. In spoken word recognition, for example, the IA-based TRACE model accounts for approximately 30 phenomena (Magnuson & Crinnion, in press).

In mathematical ecology, another dynamical approach that has been deeply influential is the Lotka-Volterra model (Lotka, 1920; Volterra, 1926). LV models the growth and decay of species that interact with one another, capturing this interaction in ways related to IA. As an extension of the logistic equation to n interacting species, LV has been extensively studied and several mathematical insights are available (Baigent, 2016).

Both frameworks are at heart ordinary differential equations (ODE) of first order, autonomous, and quadratic (or quasi-quadratic in the case of IA). But IA also exhibits some peculiarities that make it hard to analyze, hindering our theoretical understanding of it.

The purpose of this study is to determine whether an IA model endowed with gLV equations can be trained, and how it would compare to an IA baseline in terms of classification accuracy and robustness on a simple word recognition task.

2 Interactive Activation and the Generalized Lotka-Volterra model

Here we describe the Interactive Activation equations, as introduced by McClelland and Rumelhart (1981). We explain why they cannot be analyzed with extant mathematical tools. We then present the Generalized Lotka-Volterra model.

2.1 The Interactive Activation model

In the IA framework, units are organized into levels and interact within and across levels by sending each other some positive or negative amount of activation according to a certain connectivity matrix M.

A defining property of IA is one of sign-symmetry: units can cooperate (positive connection) or compete with one another (negative connection), but whatever their relation, it has to be reciprocal in nature: $sign(M_{ij}) = sign(M_{ji})$ for all units i, j.

The equations of IA are given by the following map which specifies the activation, x, of a node or unit, i:

$$x_i^{t+1} = \begin{cases} (1-d_i)x_i^t + net_i^t(1-x_i^t) & \text{if } net_i^t > 0 \\ (1-d_i)x_i^t + net_i^t x_i^t & \text{if } net_i^t \leq 0 \end{cases} \tag{1}$$

where $net_i^t = \sum_{j \neq i} M_{ij} ReLu(x_j^t)$ is the net input of unit i, and $d_i > 0$ is a decay term associated with unit i.

The IA framework has an unusual feature: a unit's activation in IA depends on the sign of its net input, yielding not one but two equations. This is meant to stabilize the system. When net_i^t is negative or 0, x_i decreases to 0 to reach equilibrium. When net_i^t is strictly positive, the equilibrium value for x_i is given by:

$$x_i^* = 1 + \frac{d_i}{net_i^*} = 1 + \frac{d_i}{\sum_{j \neq i} M_{ij} ReLu(x_j^*)} \tag{2}$$

2.1.1 Limitations

The equilibrium property above is of limited use, because it does not characterize an IA equilibrium without reference to the net input, which itself involves the equilibrium value for all other units. This leads to a quadratic equation in x, with additional non-linearity because of the ReLU in the net function, making the equilibrium intractable analytically.

The change in dynamics as a function of net input also leads to difficulties. In IA, this mechanism aims at keeping activations within fixed bounds, and at ensuring that a unit will return to its resting level when not stimulated. However, it also introduces non-differentiabilities that place these equations beyond reach of analysis. In particular, the Jacobian of the system cannot be written down, which precludes us from assessing the stability of potential equilibria without numerical simulations. An issue related to stability is that the way noise propagates within the system cannot be predicted.

Finally, to our knowledge there is no efficient procedure to set the network weights in IA.

2.2 The generalized Lotka-Volterra model

The gLV model has a distinguished history. Malthus (Malthus, 1798) expounded and formalized the results of Linnaeus (Linnaeus, 1758), that unchecked population reproduction must bring exponential growth. This famously prompted

Darwin to estimate (not quite rigorously) that even the slowest reproducing animal species on the planet, the elephant, would reach a population of 34,584,256 in 790 years of unbounded reproduction (Darwin, 1859). Asked to account for resource limitations, Verhulst introduced the logistic map (Verhulst, 1838), which has had enormous impact in mathematics as the simplest dynamical system exhibiting chaos, and whose solution (the logistic function) was later to enjoy much success in neural networks. But a model of population growth, able to acknowledge resource limitations as well as cooperation or competition between species, would have to wait for Lotka and Volterra (Lotka, 1920; Volterra, 1926). We now describe the Lotka-Volterra model for n species, gLV, which is a staple of theoretical ecology and perhaps most influential today in the study of the microbiome.

The gLV map for n populations is given by:

$$x_i^{t+1} = x_i^t(r_i + \sum_{j=1}^{n} m_{ij} x_j^t) \tag{3}$$

or in matrix form:

$$x^{t+1} = D(x^t)(r + Mx^t) \tag{4}$$

where:
$r = (r_1, r_2, .., r_n)$ is the growth rate for each species/unit;
$x^t = (x_1^t, x_2^t, .., x_n^t)$ is the density of each species at time t;
$D(x)$ is the matrix that has x on the diagonal;
M is the nxn interaction matrix.

In what follows, we will use the terms "interaction matrix" and "connectivity matrix" interchangeably, and neither will we make a distinction between the words "fixed-point", "equilibrium" and "steady-state". We also work with the continuous version of gLV, dropping time indices:

$$\frac{dx}{dt} = D(x)(r + Mx) \tag{5}$$

The gLV system can exhibit complex dynamics, including limit cycles, chaotic behavior, and some other non-equilibrium attractors. Indeed one cannot in general predict the full trajectory of the model given only its initial conditions in any other way than numerically. In almost all regimes, the dynamics of gLV allow for several fixed points, and mapping out the basins of attraction of such fixed points remains a numerical exercise. However, several formal results have been obtained and some specific cases are tractable. When M is invertible, a fixed point of the system can be written as:

$$x^* = -rM^{-1} \tag{6}$$

It is well-known of the theoretecial ecologists that if this equilibrium is positive (i.e. a "feasible" equilibrium where all components are positive), it is also unique. Under some very specific conditions (if M is Lyapunov diagonally stable) it can be shown that all positive initial states must necessarily converge to the feasible equilibrium.

Perhaps more relevant to the cognitive scientist is the finding that when there is no feasible equilibrium, all bounded trajectories are guaranteed to lead to equilibria with some extinct species (Allesina, 2020). As stated in Jones, Shankin-

Clarke, and Carlson (2020), "A gLV system with n populations can exhibit up to 2^n steady states, where each steady state is specific to a distinct presence/absence combination of the n species". This has relevance to the phenomena covered by IA, where we are typically interested in a few units being activated together and the others converging to zero (i.e. "extinct species").

3 Training dynamical systems

The idea to train ordinary differential equations is not new, but to be successful one must rise above two obstacles: computing error gradients through continuous dynamics, and potential runaway activation during training and inference –controlling the stability of the system.

3.1 Computing gradients through the dynamics

The field has matured significantly in the late 2010s with the introduction of neural ODEs (Chen, Rubanova, Bettencourt, & Duvenaud, 2018), which allow for efficient gradient-based optimization to learn ODE parameters directly from data.

Consider an ODE of the form $\frac{dx}{dt} = f(x,t,\theta)$, where $x(t)$ is the state, f is a function (often a neural network) with parameters θ, and the initial condition $x(0)$ is the input data. In the case of the gLV system, the function $f(x,t,\theta)$ would be defined as

$$f(x,t,\theta) = D(x)(r + Ax),$$

and $\theta = \{A, r\}$ would be the learnable parameters. The goal is to adjust θ so that the solution $x(t)$ at a specific time, or its steady state, aligns with a target output. This is achieved by defining a loss function L (e.g., cross-entropy for classification) and computing gradients $\frac{\partial L}{\partial \theta}$. Unlike discrete recurrent models, whose training involves computing gradients across distinct time steps, ODEs require backpropagation through continuous dynamics. The adjoint method, introduced by Chen et al. (2018), elegantly handles this by defining an adjoint variable $a(t) = \frac{\partial L}{\partial x(t)}$, which measures the sensitivity of the loss to the state, and solving a second ODE backward in time, $\frac{da}{dt} = -a(t)^T \frac{\partial f}{\partial x}$, avoiding the need to store intermediate states and making training memory-efficient.

3.2 Stability and ways to achieve it

Another key aspect to consider is the stability of the dynamical system. To ensure stability, neural ODEs usually involve a soft-bounding function such as the sigmoid, tanh, or a variant of those, that squashes activations within some bounds. But this is not the case in the gLV, and neither can we use a built-in mechanism to prevent runaway activations like IA's dual equations. Instead, in gLV, the stability of the system hinges entirely on the properties of the interaction matrix A and of the growth rate vector r.

In the study of dynamical systems, a matrix M is called stable if all the real parts of its eigenvalues are negative. This suggests that we should control these quantities during training. When training gLIA, we could attempt to keep it stable by regularizing or constraining the loss with an interaction matrix stability condition. For instance, we could build a penalty term by summing up the positive real parts of eigenvalues, or we could optimize directly under the constraint that each real part is negative, using constrained optimization techniques.

4 Methods

Our training approach follows the adjoint method of (Chen et al., 2018).

In the remainder of this article, we will call ”gLIA” any model whose state evolves according to equation 5 (gLV), and whose interaction matrix conforms to the sign-symmetry constraint (IA):

$$sign(m_{ij}) = sign(m_{ji}) \tag{7}$$

For simplicity, we consider a gLIA model with only 2 levels: sublexical units (e.g. letters) and lexical units (words). This is meant to represent a simple lexical system, as in the original IA study. The model uses localist/one-hot units: at the letter level, units represent letters at a given position in a word, whereas at the word level, each unit represents a word. Hence, word TAB is represented as T1, A2, B3, TAB.

We will use the following criterion for word recognition: a word is considered recognized if upon presentation of its constituent letters, its corresponding unit is the most active after 50 evaluations of the differential equation (i.e. time steps).

As a lexicon, we use the ”google_10000_english” dataset of the 10 000 most frequent English words, varying the number of words depending on simulations, and admitting only words of length 3 or more.

4.1 Supervised classification with the adjoint method

Let us now train gLIA as a dynamical classifier. For this, we build a dataset by encoding words as vectors of letter/positions, perturbing the input letter vectors with Gaussian noise (producing 100 noisy samples per word), and encoding target words as one-hot units. The dataset is split into train and test sets.

We consider 4 models for training: gLIA, gLIA symmetric, gLIA negative definite and IA (our baseline). The model is always initialized with the IA weight matrix template, whereby words are positively connected to their constitutent letters and vice versa, but without initial competition between words or between letter units at the same position.

During training, we present constituent letters as input to the letter units, all other units being set to a near-zero value. We then run the gLV dynamics, read the final activations in the word units and compare those to one-hot target words.

We use MSE as the loss, since intriguingly our attempts with cross-entropy, much more typical for classification, were not successful. In the symmetric condition, we also found it necessary to decrease the learning rate as a function of number of words in the model, with learning rates of 0.001 below 200 words, 0.0001 for 200 words, and 0.00001 above that. Setting the learning rate too high produced NaN errors because of runaway activations, presumably because the larger number of weight updates resulting in overly strong weights.

We train the model with Adam as optimizer, and the sum of positive real-parts of eigenvalues for regularization. This regularization term decreases during training, guiding M towards stability.

Crucially, we use the torchdiffeq library (Chen, 2018) that efficiently computes loss gradients through the gLV differential equation, using Chen et al.’s adjoint method.

5 Results

The test accuracies of the trained models in all four conditions are shown in Table 1, along with other model metrics.

Table 1 shows that the classification-trained gLIA networks can recognize words, in the sense that presenting only the constituent letters of a word at the right locations will activate the correct word above the others at the end of the run. The model can be scaled up to a large number of words, even with a very naive ”slot coding” scheme. Performance

Table 1: Performance metrics for different model sizes and configurations. Note: For "gLIA Sym." and "gLIA Neg. Def.", the number of parameters in the symmetric case is calculated as $n(n+1)/2$, where n is the number of units.

Words	Samples	Epochs	Units	Parameters	Parameters (Sym)	Test Acc. (%)			
						gLIA	gLIA Sym.	gLIA Neg. Def.	IA
10	1,000	200	114	12,996	6,555	100.00	100.00	100.00	100.00
20	2,000	200	144	20,736	10,440	100.00	100.00	98.87	100.00
30	3,000	200	160	25,600	12,880	100.00	100.00	96.67	96.33
100	10,000	200	181	32,761	16,471	100.00	100.00	100.00	98.67
200	20,000	200	310	96,100	48,205	86.17	94.10	90.28	74.67
500	500,000	200	708	501,264	251,586	36.94	89.88	89.74	5.66
1000	1,000,000	200	1157	1,338,149	669,903	20.41	91.69	90.34	0.30

improves in the symmetric case and is at its best in the negative definite condition. Interestingly, IA can also be trained using the adjoint method, although all gLIA models outperform the IA baseline.

Another observation is that, assuming small enough learning rates are used during training, the models are stable during inference. In our simulations, for initial conditions contained within [0, 1], we always observed bounded evolutions. This is obtained despite the absence of stabilizing effects of IA's dual equations, and of course without adding artificial activation bounds. The bounded dynamics of the generic gLIA model, combined with its differentiability open up the possibility of a mathematical treatment.

5.1 Symmetric variant: global stability and a Lyapunov function

In the symmetric variant of gLIA (i.e. when we enforce symmetry of the interaction matrix at every gradient step), performance is improved compared to the non-symmetric case.

A key fact, however, is that the final trained symmetric M can be made definite negative, essentially without deteriorating classification performance. This has deep implications, as the system can then be proved to be globally stable.

Indeed, as a negative definite matrix, M is automatically Lyapunov-diagonal, implying that the system admits a Lyapunov function, which is well known to theoretical ecologists (Allesina, 2020). As it turns out, in this special case of a negative definite M, one can introduce an even more straightforward function:

$$L(x) = -r^T x - \frac{1}{2}x^T M x - \frac{1}{2}r^T M^{-1} r.$$

We show in Appendix 1 that when the interaction matrix M is negative definite, L is strictly convex, non-negative, with the equilibrium point $M^{-1}r$ as its unique zero. In other words, L is a Lyapunov function for gLV, and therefore also for gLIA. This implies that the system is bounded for all possible evolutions. Figure 1 illustrates the evolution of a few initial conditions towards states that efficiently serve word classification.

It can be seen that, as indicated by formal analysis, when the weight matrix is negative definite, Lyapunov values are strictly decreasing along the model trajectories, while still allowing for well-separated final states.

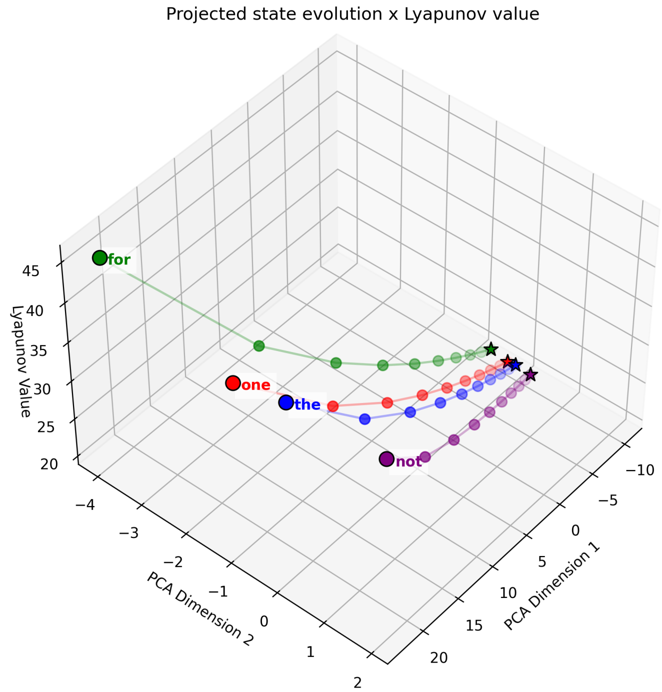

Figure 1: Projected state evolution and their Lyapunov values for selected words, in the Lyapunov-stable gLIA.

6 Discussion

We set out to determine whether the recently introduced gLIA model could be trained efficiently. Our results show that this is the case for the simple tasks that we have considered. In particular, we have showed that:

- gLIA models can be trained that recognize upwards to 900 English words, with a training method requiring minimal hyper-parameter tuning.
- The asymmetric gLIA model has bounded dynamics, and yet is fully differentiable, contrary to IA.
- The symmetric gLIA model obtains superior performance and can be made Lyapunov-stable.

6.0.1 Trainable parameters

In this work we have not considered the growth rates as trainable parameters: they were set to a constant and fixed value for all units. Growth rates also do not play a role in the proof of Lyapunov stability for negative definite weight matrices. Nonetheless, training growth rates along with the interaction matrix could improve performance. In the simple two-layer language gLIA model that we have considered, growth rates could also be advantageously initialized with values proportional to the logarithm of letter and word frequencies.

6.0.2 Scaling network units

NeuralODEs were initially introduced as the continuous limit to the sequential updates undergone by hidden states in discrete network layers such as Residual Networks (Chen et al., 2018). However, it is also clear that one could encode arbitrarily deep multi-layered networks into the interaction matrix of a gLIA model, as long as one accepts the gLV activation update for all neurons across all layers. While the interaction matrix scales like the square of the number of units, which makes direct computation quickly unfeasible for large dimensions, this is in fact not an issue because in a multi-layer network the matrix would also be very sparse, with only a sequence of blocks along its diagonal needing updates.

Alternatively, gLIA layers can be introduced before, after or in-between the neural network layers used in modern deep learning, like convolution, pooling and attention layers. Such a use of NeuralODE's as components in more complex networks has been demonstrated many times, starting from the original paper of Chen et al. (2018).

6.0.3 Are words attractors of the gLIA network?

Let us recall that in gLV and gLIA, when all trajectories are bounded and the global equilibrium is not feasible, they must eventually converge to attractor states with ”extinct species”. In our simulations, the equilibrium is never feasible, and it is thus natural to ask whether gLIA represents words as such equilibria, with all units showing zero activation (”extinct”) except for the ones corresponding to the word and its constituent letters.

This question can be easily decided by running the model from the initial state of interest, plugging in the end state into the equation and determining whether the outcome vector update is zero, i.e. whether the state is a fixed point. If that's the case, unlike in IA, one can then evaluate the stability of fixed points in gLIA, owing to a simple expression of the Jacobian matrix.

In our simulations however, we observe that the final states have not converged. Therefore, rather than being fixed points of the dynamical system, words in gLIA are encoded as regions of sufficiently slow evolution of the equation. In other words, the signal drives the model towards a region where its state is stable but distinct enough from other states to serve classification.

6.0.4 Sequences of stimuli

Finally, throughout this article, our setup for studying IA and gLV has been that of ODEs. But the study of IA dynamics with time-varying inputs, particularly important in sequential processes, would appear to call for control theory. At least one study (Jones et al., 2020) has reported results on gLV when augmented with control, we expect that further research in this direction could bring more insights into gLIA as a model of cognition.

References

Allesina, S. (2020). *A tour of the generalized lotka-volterra model.* Retrieved from `https://stefanoallesina.github.io/Sao`$_paulo_school/$

Baigent, S. A. (2016). Lotka-volterra dynamics: An introduction. *World Scientific*.

Chen, T. Q. (2018). *torchdiffeq.* Retrieved from `https://github.com/rtqichen/torchdiffeq`

Chen, T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, *31*, 6571–6583.

Darwin, C. (1859). *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. John Murray.

Jones, J., Shankin-Clarke, P., & Carlson, J. (2020). Navigation and control of outcomes in a generalized lotka-volterra model of the microbiome. *AIMS Series on Applied Mathematics: Advances in Nonlinear Biological Systems: Modeling and Optimal Control*, *11*(7), 97–120.

Linnaeus, C. (1758). *Systema naturae*. Holmiae (Stockholm): Laurentii Salvii.

Lotka, A. (1920). Analytical note on certain rhythmic relations in organic systems. *Proceedings of the National Academy of Sciences*, *6*(7), 410–414.

Magnuson, J. S., & Crinnion, A. M. (in press). Spoken word recognition. In J. Trueswell, A. Papafragou, & L. Gleitman (Eds.), *Handbook of the Mental Lexicon.*

Malthus, T. R. (1798). *An essay on the principle of population*. Library of Economics, Liberty Fund, Inc. (Available at: `https://www.econlib.org/library/Malthus/malPop.html`)

McClelland, J. L., Mirman, D., Bolger, D. J., & Khaitan, P. (2014). Interactive activation and mutual constraint satisfaction in perception and cognition. *Cognitive Science*, *6*, 1139-1189. doi: 10.1111/cogs.12146

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: I. An account of basic findings. *Psychological Review*, *88*(5), 375–407. doi: 10.1037/0033-295X.88.5.375

Verhulst, P.-F. (1838). Notice sur la loi que la population suit dans son accroissement. *Correspondance mathématique et physique*, *10*, 113-121.

Volterra, V. (1926). Fluctuations in the abundance of a species considered mathematically. *Nature*, *118*, 558–560.

A Lyapunov Function for the Generalized Lotka-Volterra Model

A.1 Dynamics and Lyapunov Function

Recall the generalized Lotka-Volterra (gLV) differential equation:

$$\frac{dx}{dt} = D(x)(r + Mx),$$

where $D(x)$ is a diagonal matrix with x on its diagonal, $r \in R^n$ is a vector of intrinsic growth rates, and $M \in R^{n\times n}$ is the interaction matrix. If x is initialized in the positive orthant ($x(0) > 0$), the dynamics ensure that $x(t)$ remains in the positive orthant for all $t \geq 0$.

We introduce the following scalar-valued function:

$$L(x) = -r^T x - \frac{1}{2} x^T Mx - \frac{1}{2} r^T M^{-1} r.$$

We will now show that if the interaction matrix M is negative definite, L satisfies the conditions for a Lyapunov function: it is non-negative, monotonically decreasing under the dynamics of gLV, and bounded below by zero. To show this, we first establish that L is convex.

A.2 Convexity of L

The function $L(x)$ is strictly convex. To prove this, we first compute its gradient before obtaining its Hessian.

The gradient of $L(x)$ is given by:

$$\nabla L(x) = -r - Mx.$$

The Hessian of $L(x)$ is the matrix of second partial derivatives. Taking the gradient of $\nabla L(x)$, we obtain:

$$\nabla^2 L(x) = \frac{\partial}{\partial x}(-r - Mx) = -M.$$

$$\nabla^2 L(x) = -M.$$

Since M is negative definite, $-M$ is positive definite. Thus, the Hessian $\nabla^2 L(x)$ is positive definite, and $L(x)$ is strictly convex. This guarantees that $L(x)$ has a unique global minimum at x^*.

A.3 Non-Negativity of L

For M negative definite, we have $L(x) \geq 0$ for all $x \in R^n_+$. The minimum of $L(x)$ occurs at the equilibrium point $x^* = -M^{-1}r$, where $L(x^*) = 0$.

To prove $L(x) \geq 0$, we find the critical point x^* of $L(x)$ by setting the gradient to zero:

$$\nabla L(x) = -r - Mx = 0 \implies x^* = -M^{-1} r.$$

Substituting x^* into $L(x)$:

$$L(x^*) = -r^T(-M^{-1}r) - \frac{1}{2}(-M^{-1}r)^T M(-M^{-1}r) - \frac{1}{2} r^T M^{-1} r.$$

Simplifying:

$$L(x^*) = r^T M^{-1} r - \frac{1}{2} r^T M^{-1} r - \frac{1}{2} r^T M^{-1} r = 0.$$

As established above, since M is negative definite, $L(x)$ is a convex function, and x^* is its global minimum. Thus, $L(x) \geq L(x^*) = 0$ for all x.

A.4 Monotonicity of L

The monotonicity of $L(x)$ is already implied by its convexity, but it can also be shown by noticing that its time derivative is non-positive, i.e. the function $L(x)$ is non-increasing along trajectories of the gLV dynamics. The time derivative of $L(x)$ is:

$$\frac{dL}{dt} = \nabla L(x)^T \frac{dx}{dt} = (-r - Mx)^T D(x)(r + Mx).$$

Simplifying, we obtain:

$$\frac{dL}{dt} = -(r + Mx)^T D(x)(r + Mx) \leq 0,$$

since $D(x)$ is positive definite for $x > 0$. Thus, $L(x)$ is monotonically decreasing and a valid Lyapunov function for the gLV system.