REU 2022: UW-Madison

Goyal, Agam
agoyal25@wisc.edu

Xu, Bryan
zxu455@wisc.edu

Wu, Bella
zwu363@wisc.edu

Chen, Binhao
bchen276@wisc.edu

July 8, 2022

1 Annotated Bibliography

1.1 Synchronization of finite-state pulse-coupled oscillators[5]

In this article, the basic setting of Firefly Automata (FCA) was introduced, along with the conditions for synchronization at various graph structures and initial phases. To begin with, the updating rule of FCA should be illustrated. Assume we are having a finite simple graph $G = (V, E)$. The number of nodes is denoted by n. The map X maps the set of vertices to a corresponding state, written as $X : V \to Z/nZ$, where the later is a cyclic group taking order $0 < 1 < \cdots < n-1$. So at a specific time, the node v takes the value of $X_t(v)$. Define the neighbors of blinking states as a set $N(b)$, the blinking state as $b(n) = \lfloor \frac{n-1}{2} \rfloor$, then the transition rule of FCA writes:

$$X_{t+1}(v) = \begin{cases} X_t(v) & X_t(v) \in N(b), X_t(v) > b(n) \\ X_t(v) + 1 & Else \end{cases}$$

The theorems related to it show some necessary and sufficient conditions for the configuration to synchronize for every initial configuration when having $n \in \{3, 4, 5, 6\}$. And the key point is that each vertex should be able to blink infinitely times and not pulled from updating. Not surprisingly, the vertex that is being dragged is the one that has higher degree than n. Therefore, all specific nodes that has lower degrees can blink infinitely and not impeded. What naturally follows is that if the maximum degree of the tree is strictly less than n, then it synchronizes. The reverse direction is the easier part. The key take away of the content of this paper related to FCA is the insight that some of the configurations or settings are guaranteed to synchronize under some conditions. Further in this direction, for graphs with m vertices which are destined to have n-synchronization, there exists a maximum synchronizing time given as:

$$n(\frac{n}{2} - 1 + m) \leq T_n(m) \leq (m-1)(\frac{n^2}{2} + 2n - 2)$$

1.2 Phase Transition In Firefly Cellular Automata on Finite Trees[4]

In 1.1, it was shown that for any $k \geq 3$, an arbitrary $k-$coloring on finite paths synchronizes in finite time. Moreover, further work showed that even for infinite paths, the probability of attaining synchronization on a finite interval t approaches 1.

However, this result does not carry forward to finite trees, where this universal synchronization behavior isn't observed for $k \geq 3$.

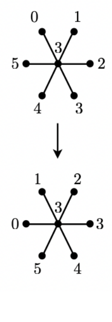

Figure 1: 6-color non-synchronizing example on a Finite Tree

As we see in Figure 1, the central vertex of the tree, which has current state of 3, is stuck in a situation where the blinking color is $b(k) = \lfloor \frac{n-1}{2} \rfloor = 2$, and consequently, it can never *pull* it's neighboring nodes and thus always stays in the same state and prevents the eventual synchronization of the graph.

More generally, let us consider a vertex v with degree $\geq k$ of a finite tree T, and let $T_1, \cdots, T_m$ be the connected components of $T - v$ which is the graph we obtain from T by deleting v as well as all edges incident to it. Next, we assign a state $i\ (mod\ k)$ to every vertex of T_i, and assign any state $> k/2$ to vertex v. Then like we observe in Figure 1, v never blinks and each component T_i will never get pulled by v.

The key takeaway from this paper is to extend the results about synchronization for $k \in \{3, 4, 5\}$ from section 1.1, to all general k as follows:

(i) If $k \in \{3, 4, 5, 6\}$ and $T = (V, E)$ is any finite tree, then every $k-$coloring on T synchronizes iff T has maximum degree $< k$.

(ii) If $k \geq 7$, then there exists a finite tree $T = (V, E)$ with maximum degree $\geq k/2 + 1$ and a *non-synchronizing* $k-$coloring on T.

1.3 Learning to predict synchronization of coupled oscillators on heterogeneous graphs[1]

As illustrated in the title of the paper, heterogeneous graphs are the main interest. Heterogeneous graphs corresponds to the set of non-isomorphic graphs. The definition of isomorphic graphs is illustrated by the same structure in both nodes and connection topology of the graphs. Two graphs are called isomorphic if the following holds true:

$$\exists\, \Phi : V(G) \rightarrow V(H)$$

$$v_1 v_2 \in A(G) \iff \Phi(v_1)\Phi(v_2) \in A(H)$$

where Φ is a bijective mapping from vertices in a graph G to one of its isomorphic graphs H. We define A to be set of adjacent pairs of nodes that are connected by a single line segment. Thus the second line says a necessary condition for isomorphism is that if two vertices are adjacent, then so do their corresponding vertices in the isomorphic graph, vice versa. In the setup discussed in the paper, the non-isomorphic property is mainly the failure of the second condition, where we have bijection between nodes and same sequence of order of nodes. As nodes of the graph grows gradually, the number of heterogeneous graphs explodes and the system rapidly becomes intractable.

Therefore, the philosophy behind our project is to treat this analytically intractable problem as a binary classification on whether a graph is synchronizing or not. Main features of the machine learning algorithm are the initial dynamics. The main learning algorithms that will be used here are Random Forest Gradient Boosting (RFGB), Feed Forward Neural Networks (FFNN), and Long-term Recurrent Convolutional Networks (LRCN).

The three physical models that are being evaluated are Kuramoto (KM), Firefly Automata (FCA), and Greenberg-Hasting (GHM). KM is a differential equation that considers the sine function of difference between states of neighboring points with certain coupling strength κ and is numerically carried out by discretizing time steps. FCA is introduced in previous section, or refer to [Lyu15] & [Lyu16]. GHM is another discrete model having the following updating rules:

$$\mathbf{X}_{t+1}(v) = \begin{cases} 0 & \mathbf{X}_t(v) = 0, \mathbf{X}_t(u) \neq 1\, \forall u \in N(v) \\ 1 & \mathbf{X}_t(v) = 0, \mathbf{X}_t(u) = 1\, for\, some\, u \in N(v) \\ X_t(v) + 1 & Else \end{cases}$$

However, even the three models are initialized through the same configuration $\mathbf{X}_0$, they follow distinct evolutionary paths which impedes a universal method of predicting synchronization. Before going into the machine learning predictor, a baseline predictor routed from analytical proof will be introduced as a reference. It states that the phase at any time is confined in an open half circle in the phase space Ω. The baseline predictor is conservative due to the fact that it only predicts synchronization when the concentration principle is observed and flip a fair coin otherwise.

Since L2PSync could successfully predict the synchronization of an unknown system with surprising accuracy, this paper proposed two hypothesis. First, during the initial period, the entropy of the system would decay rapidly until the uncertainty of the future behavior from unknown graph structure becomes insignificant. Second, there might be a more generalized concentration principle that uses the whole initial dynamics. After computing the Gini index of all features, the paper found that full graph information does not improve the prediction accuracy compared with models using only initial dynamics. Finally, L2PSync is robust on large systems even if only partial initial dynamics information.

1.4 Non-negative Matrix Factorization[2]

Non-negative matrix factorization (NMF) is an algorithm devised to learn part-based representations of whole objects rather than focusing on a holistic figure as in Principal Component Analysis (PCA) or Vector Quantization (VQ). So in a task of learning facial objects - while the basis images for PCA would be "eigenfaces" which represent distorted versions of the entire face, for NMF the basis images would be "parts of the whole face" that are combined together in an *additive* fashion to obtain the full image. Sparseness in both basis and encoding of NMF is crucial for part-based representation. NMF can also be applied to semantic analysis of text documents, since this method is able to group semantically-related words into semantic features. However, NMF of single level is not specialized in learning parts from any database with great complexity compared with models with multiple levels of hidden variables.

In NMF, if we have a data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ and a rank $r \in \mathbb{N}$, we define a *dictionary* $\mathbf{W} \in \mathbb{R}^{d\times r}$ and an *encoding* $\mathbf{H} \in \mathbb{R}^{r\times n}$, and solve the following optimization problem under the L2-norm (Frobenius) or a more generalized approach under the KL divergence loss

$$\inf_{\mathbf{W}\in\mathbb{R}^{d\times r}_{\geq 0},\mathbf{H}\in\mathbb{R}^{r\times n}_{\geq 0}} \|\mathbf{X} - \mathbf{W}\mathbf{H}\|^2$$

with non-negativity constraints on both $\mathbf{W}$ and $\mathbf{H}$.

This problem is then optimized using *Alternating Least Squares* (ALS). In our work, we will use this as a technique for supervised dictionary learning to be able to perform the classification task and understand the relevant features that our model is learning, especially because of its high interpretability.

1.5 Supervised Dictionary Learning[8]

Dictionary Learning is a machine learning technique that is used to learn interpretable latent structures of complex data sets in order to realize what features of the data the model considers to be the most relevant ones for its task. It consists of two main tasks: (1) Sampling a large number of structured subsets (usually square patches) of a data set, and (2) Applying NMF as described in section 1.4, to find a set of basis elements that form our *dictionary*. Each element of this learned dictionary describes a latent shape in the image.

Supervised Dictionary Learning (SDL) aims to also include the task of classification using these learned dictionaries along with class-labels in a manner that the dictionaries learned not just perform their original task of *reconstruction*, but also perform a *discrimination* task that aids classification.

For our work, since we're looking to predict 'synchronization' or 'non-synchronization' apart from just learning what latent representations the ML algorithms are learning we will be using an SDL task. Let us consider n labeled signals (x_i, y_i) for $i = 1, \cdots, n$, where $x_i \in \mathrm{R}^p$ is a $p-$dimensional signal and $y_i \in \{0, 1\}$ is the binary label we're looking to classify our graphs into. Since dictionary learning and the classification task may not necessarily have aligned objectives, we look to optimize the SDL task using a trade-off parameter ξ to control what task is the most important to us, best representation learning or best classification accuracy.

1.6 Online Learning for Matrix Factorization and Sparse Coding[7]

This paper proposed an online dictionary learning algorithm, which is based on stochastic approximations, processing one sample at a time or using mini-batch. This iterative optimization algorithm minimizes a quadratic local surrogate function of the expected cost at each step, ensuring the efficiency of the algorithm and making it scalable to large datasets. The convergence analysis demonstrates that the algorithm converges almost surely to the stationary point of the objective function. Moreover, with second-order information of the cost function, this algorithm removes the problem of tuning the learning rate. The extension of this approach includes matrix factorization formulations and sparse principle component analysis. After experimenting with natural images and genomic data, this approach requires less computational time and is suitable for image reconstruction tasks.

1.7 From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators[9]

This is a review on the successes and failures of the 25-year research on Kuramoto model, which was motivated by the phenomenon of collective synchronization. Kuramoto first reduced Winfree's phase model to a set of governing equations, making it easier to analyze, and he later rewrote the governing equations in terms of order parameter, setting up a positive feedback loop between coupling and coherence. In his earliest work, he proved that the group of oscillators would split into two groups, "locked" and "drifting" in the long term. Nevertheless, there were two major issues left unresolved in Kuramoto's analysis—finite-N fluctuations and the stability of the steady solutions. Kuramoto and Nishikawa proposed two pioneering stability theories, but neither turned out to be correct. Later, the author came up with a contiuum limit of the Kuramoto model by phrasing the continuum limit as densities, able to address existence, stability, and bifurcation of various solutions. Moreover, Sakaguchi extended the Kuramoto model to allow rapid stochastic fluctuations in the natural frequencies, and meanwhile, the author solved the linear stability problem for the incoherent state of Sakaguchi's model. Recently, Crawford made significant contributions to the analysis of Kuramoto model, including the first systematic formulation of the weakly nonlinear stability problem for the incoherent state, and so on. Crawford also implied that instabilities are created by unstable modes emerging from a continuous spectrum.

1.8 Task-Driven Dictionary Learning[6]

By tuning the dictionary to specific tasks, task-driven dictionary learning is different from the classical data-driven approach to dictionary learning, which only adapts the dictionary to data reconstruction. First, the paper introduced a basic supervised formulation for regression or classification tasks, able to handle large-scale data. Under the supervised setting, one thing to note is that it is not necessary to have more elements in dictionaries than the signal dimension, since perfect reconstruction is not required as long as discriminative features are captured. Next, the paper presented two extensions to the previous formulation: learning a linear transform of the input data and semi-supervised learning. Applications include regression, binary classification, multiclass classification, and compressed sensing.

1.9 Supervised dictionary learning with auxiliary covariates[3]

This paper gave a systematic study of supervised dictionary learning (SDL) and investigated the incorporation of auxiliary information to SDL model to improve classification accuracy. For weakly constrained SDL, it is possible to 'lift' the original nonconvex SDL problem into a convex problem in a larger space with low-rank constraints. For strongly constrained SDL, an efficient block coordinate descent (BCD) algorithm for SDL is provided to find an -stationary point of the objective. Furthermore, this paper explored the stability difference between feature-based SDL models (SDL-feat) and filter-based SDL models (SDL-filt). Since SDL-feat makes use of the code matrix, in classification tasks, the correct code representation has to be derived in supervised learning. In comparison, the prediction pipeline for SDL-filt is more streamlined, and furthermore, SDL-filt have been studied extensively in the supervised matrix factorization literature. In terms of convergence behavior, SDL-filt is more stable than SDL-feat, even though SDL-feat converges faster than SDL-filt when working with large problem dimension. This is due to the fact that SDL-filt converges to the global optimum without any additional L2-regularization, while SDL-feat requires L2-regularization despite its fast computational time.

References

[1] Hardeep Bassi et al. "Learning to predict synchronization of coupled oscillators on heterogeneous graphs". In: *arXiv preprint arXiv:2012.14048* (2020).

[2] Daniel D Lee and H Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791.

[3] Joowon Lee, Hanbaek Lyu, and Weixin Yao. "Supervised Dictionary Learning with Auxiliary Covariates". In: *arXiv preprint arXiv:2206.06774* (2022).

[4] Hanbaek Lyu. "Phase transition in firefly cellular automata on finite trees". In: *arXiv preprint arXiv:1610.00837* (2016).

[5] Hanbaek Lyu. "Synchronization of finite-state pulse-coupled oscillators". In: *Physica D: Nonlinear Phenomena* 303 (2015), pp. 28–38.

[6] Julien Mairal, Francis Bach, and Jean Ponce. "Task-Driven Dictionary Learning". In: *IEEE transactions on pattern analysis and machine intelligence* 34 (2011).

[7] Julien Mairal et al. "Online Learning for Matrix Factorization and Sparse Coding". In: *Journal of Machine Learning Research* 11 (2010).

[8] Julien Mairal et al. "Supervised dictionary learning". In: *Advances in neural information processing systems* 21 (2008).

[9] Steven H Strogatz. "From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators". In: *Physica D: Nonlinear Phenomena* 143 (2000).