

# Label Propagation Through Neuronal Synchrony

Marcos G. Quiles, Liang Zhao, Fabricio A. Breve, and Anderson Rocha

**Abstract**—Semi-Supervised Learning (SSL) is a machine learning research area aiming at the development of techniques which are able to take advantage from both labeled and unlabeled samples. Additionally, most of the times where SSL techniques can be deployed, only a small portion of samples in the data set is labeled. To deal with such situations in a straightforward fashion, in this paper we introduce a semi-supervised learning approach based on neuronal synchrony in a network of coupled integrate-and-fire neurons. For that, we represent the input data set as a graph and model each of its nodes by an integrate-and-fire neuron. Thereafter, we propagate the class labels from the seed samples to unlabeled samples through the graph by means of the emerging synchronization dynamics. Experiments on synthetic and real data show that the introduced technique achieves good classification results regardless the feature space distribution or geometrical shape.

## I. INTRODUCTION

In our digital age, information reaches us at remarkable speed and the amount of data it brings is unprecedented. In the hope of understanding such flood of information, data mining and machine learning approaches are required. In this scenario, one of the problems we often face is data categorization (classification).

There are several approaches for data categorization in the literature and one of the problems they need to address, in almost every situation, is how to obtain labeled data to feed the categorization systems in order to achieve operational results. Notwithstanding, the labeling process of acquired data is costly, time-consuming, and usually requires expert knowledge. For these reasons, it is not uncommon to have only a small subset of labeled data at hands.

Traditional classifiers are constructed based on supervised learning; then, only labeled data are considered for the training process and unlabeled examples are simply ignored. On the other hand, clustering techniques perform unsupervised learning, where label information is ignored. To fill this gap, semi-supervised classifiers are designed to learn from both labeled and unlabeled data, becoming a new topic of research that has received increasing attention in the past few years [1]–[3].

Marcos G. Quiles is with the Department of Science and Technology (DCT), Federal University of São Paulo (UNIFESP), São José dos Campos, SP, Brazil, (phone: +55 12 3942-5568; e-mail: quiles@unifesp.br). Part of this work was undertaken while M. G. Quiles was a postdoctoral researcher supported by FAPESP at the Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), São Carlos, SP, Brazil.

Liang Zhao and Fabricio A. Breve are with the Department of Computer Science, Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP), São Carlos, SP, Brazil (e-mail: {zhao, fabricio}@icmc.usp.br).

Anderson Rocha is with the *Reasoning for Complex Data* (RECOD) Lab., Institute of Computing (IC), University of Campinas (UNICAMP), Campinas, SP, Brazil (e-mail: anderson.rocha@ic.unicamp.br).

Some semi-supervised methods available in the state-of-the-art include: generative models [4], [5]; cluster-and-label techniques [6], [7]; co-training and tri-training techniques [8]–[11]; low-density separation models, such as Transductive Support Vector Machines (TSVM) [12]; and graph-based methods, such as Mincut [13], Local and Global Consistency [14], and label propagation techniques [15], [16]. Amidst all these approaches, graph-based methods are the most active research topic in semi-supervised learning [2].

Most graph-based methods can be viewed as estimating a function on the graph that satisfies two conditions: closeness to the given labels and smoothness over the graph [1]. This is equivalent to a regularization framework, and most methods are very similar differing only in their choices of loss functions and regularizers [13], [14], [17]–[20]. In addition, most methods are transductive and a relearning process is needed when new data samples are added. Thus, the development of alternative semi-supervised learning techniques by introducing dynamics into the network is undoubtedly important.

An interesting self-organization phenomenon observed in several coupled systems in nature is the mutual synchronization of coupled oscillators [21]. The synchronization has been applied to solve several problems. In special, models based on the oscillatory correlation framework proposed in [22] have been developed and applied to several domains, such as: image segmentation [23], [24], speech segregation [25], data clustering [26], visual and auditory selective attention [27]–[29], among others. See [30] for an extensive review. Basically, the oscillatory correlation theory describes that each object/cluster is represented by an assembly of oscillators with synchronous activity whereas distinct objects are represented by desynchronized groups of oscillators.

Although some work have been done on developing oscillatory correlation models to machine learning problems (e.g., [26]), the use of synchronization of neurons have not yet been applied to semi-supervised learning problems. In this direction, we propose a new method for semi-supervised classification based on neuronal synchrony. We use the local synchrony between neurons as an alternative way of propagating labels in the network.

The motivation behind the study of semi-supervised learning techniques is twofold. First, it is extremely useful in several practical situations, in particular when the labeling process is expensive; second, it has theoretical importance in understanding learning processes by machine and biological systems. For instance, it has been argued that the temporal correlation could be seen as a mechanism to bind together different images of the same object into a perceptual whole.

This process is directly related to semi-supervised learning, in which the label associated to one of those images is propagated to the whole object allowing humans to recognize, for example, a particular face viewed from an angle/position not previously observed [31].

Finally, we believe our approach has several applications. We can use label propagation in content-based retrieval systems [32] as well as in supervised multi-classification problems [33]–[36] when only a limited amount of labeled data is available.

The rest of the paper is organized as follows. Section 2 describes our model. Section 3 presents experiments and results. Finally, Section 4 draws some conclusions and discusses some future directions.

## II. MODEL DESCRIPTION

Basically, semi-supervised learning consists of a training process for a given data set in which just a small portion of the samples is labeled (annotated). In this paper, we present a new graph-based dynamic method for label propagation of unlabeled data points. For this purpose, the input data is firstly transformed into a graph (network); then, each node of the graph is modeled as an integrate-and-fire neuron. Class labels are propagated from seed samples to unlabeled samples through the graph by means of the emerging synchronization dynamics of the network. Figure 1 depicts an overview of the proposed model.

### A. Network Formation

As a graph-based semi-supervised learning method, the first step consists of generating a network  $G = (V, E)$  for a given data set. The input data set is represented by the input samples  $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$  and a set of labels  $\mathcal{L} = \{1, 2, \dots, c\}$ , where  $n$  and  $c$  are the number of samples and classes, respectively. In the semi-supervised scenario, the input data set  $\mathcal{X}$  is divided into a labeled subset  $\mathcal{X}_l = \{x_1, x_2, \dots, x_l\}$  and an unlabeled subset  $\mathcal{X}_u = \{x_{l+1}, x_{l+2}, \dots, x_n\}$ . Typically, the number of labeled samples is much smaller than the unlabeled ones. Each sample  $x_i \in \mathcal{X}_l$  has an associated label  $y_i \in \mathcal{L}$ .

Here, nodes  $V = \{v_1, v_2, \dots, v_n\}$  represent the data samples  $\mathcal{X}$  and edges  $E$  represent similarity between them. We can represent the edges in  $E$  using an adjacency matrix  $W$  such that

$$w_{ij} = \begin{cases} 1 & \text{if } d(x_i, x_j) \leq \sigma \quad \text{and} \quad i \neq j \\ 0 & \text{if } d(x_i, x_j) > \sigma \quad \text{or} \quad i = j \end{cases} \quad (1)$$

where  $w_{ij}$  defines if there is an edge between nodes  $v_i$  and  $v_j$  corresponding to the samples  $x_i$  and  $x_j$ , respectively.  $\sigma$  is a connection threshold that defines the minimum similarity criterion where a connection between a pair of nodes can be set.  $d(x_i, x_j)$  is any distance function.

We also define a set of connections between each pair of samples in  $\mathcal{X}_l$ . Given that we have a prior knowledge about the samples in  $\mathcal{X}_l$ , we can use this information to set conditional label connections. Such connections have the purpose of linking distant points if they belong to the same

class while removing links between samples of different classes. These edges are defined as

$$\forall i \in \mathcal{X}_l \text{ and } \forall j \in \mathcal{X}_l \text{ and } i \neq j \\ w_{ij} = \begin{cases} 1 & \text{if } y_i = y_j \\ 0 & \text{if } y_i \neq y_j \end{cases} \quad (2)$$

Finally, in order to ensure the graph  $G$  is connected, a new edge linking each isolated node to its corresponding closest neighbor is added

$$w_{ik} = 1 \quad | \quad \arg \min_k d(x_i, x_k) \quad (3)$$

where node  $i$  is not connected to graph  $G$  and  $d(x_i, x_k)$  is a distance function.

### B. The Integrate-and-Fire Network

In the proposed semi-supervised learning approach, each node of the network is modeled by an *Integrate-and-Fire* (I&F) neuron [37] with two types of connections to other neurons: excitatory and inhibitory connections.

- **Excitatory connections:** represent a cooperative mechanism responsible for synchronizing a group of neighbor neurons (a cluster of neurons) and also to propagate the label between neighbors. These connections are defined by the edges of graph  $G(v, e)$  (c.f., Section II-A).
- **Inhibitory connections:** connect each neuron to a global inhibitor which creates a competitive mechanism responsible for breaking the synchrony between distinct groups of neurons.

According to the coupling strength set to each connection, we may observe different situations in the network. If the excitatory connection strength is set to a high value while the inhibitory connections assume a very low one, we will probably face a global synchronization among all neurons in the network. If the contrary situation takes place, either neurons will hardly reach a synchrony state, or this synchronization state will result in several groups with a few neurons within each one.

In this model, each I&F neuron is defined as

$$\dot{\phi}_i = -\phi_i + I_i + E_i(t) - Y_i(t), \quad (4)$$

where  $\phi_i$  refers to the phase of the neuron  $i$ ,  $I_i$  defines the external stimulus, and  $E_i(t)$  and  $Y_i(t)$  define the coupling terms.

When the potential of a neuron  $i$  reaches a threshold defined by  $\theta$  ( $\phi_i \geq \theta$ ), the neuron fires and its phase is reset to zero  $\phi_i = 0$ . In addition, the neuron's pulse is propagated to neighbor neurons to change their potential.

The excitatory coupling term  $E_i(t)$  is defined as

$$E_i(t) = \sum_{j \in \Delta_i} \omega_{ij} \delta(t - t_j) \quad (5)$$

where  $\delta$  is the delta Dirac function,  $t_j$  represents the firing time of neuron  $j$ , and  $\Delta_i$  is the cooperative neighborhood of neuron  $i$  defined by the network connections. Finally,  $\omega_{ij}$

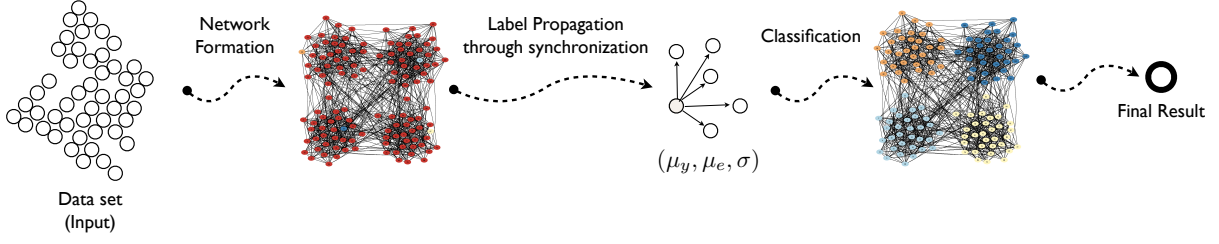


Fig. 1. Illustration diagram of the proposed model.

represents the excitatory coupling strength between neuron  $i$  and  $j$  defined as

$$\omega_{ij} = \frac{\mu_e}{|\Delta_i|}, \quad (6)$$

where  $\mu_e \in [0, 1]$  is a parameter associated to the excitatory coupling strength and  $|\Delta_i|$  represents the degree of node  $i$  in the network. The coupling normalization in Equation (6) guarantees that all neurons have the same phase trajectory by receiving the same amount of stimulus from its neighborhood [24], [38].

If  $\mu_e$  is set to a high value, the excitatory coupling strength becomes strong and neurons can easily reach a synchronous state. On the other hand, if a very low value is used, the synchrony is hardly achieved.

By analyzing Equations (4) and (5), one can see that whenever a neuron  $j$  in the neighborhood of a neuron  $i$  fires, it instantly increases the potential of neuron  $i$ . Thus, owing to the influence of a neuron over its neighbors, after a couple of cycles, neurons start pulsing in synchrony. At the same time that neighbor neurons reach synchrony, if this neuron  $j$  represents a labeled sample, the sample  $x_i$  represented by the neuron  $i$  inherits this label and it is moved from the subset  $\mathcal{X}_u$  to the subset  $\mathcal{X}_l$ . This label propagation dynamics is valid to all neurons, except those representing the initial pre-labeled samples. Those neurons have their associated labels fixed once they were assigned by a specialist and not by the technique itself. As a consequence, after a number of iterations, the neurons, through the synchrony, tend to propagate their labels over their neighborhood.

In order to avoid a global synchronization among all neurons in the network, we also consider a global inhibitor [22]

$$Y_i(t) = \frac{\mu_y}{N} \sum_{j \in \mathcal{X}_l \& y_i \neq y_j} \delta(t - t_j) \quad (7)$$

where  $\mu_y \in [0, 1]$  is a parameter that defines the inhibitory coupling strength,  $N$  represents the number of neurons (nodes) in the network and  $\mathcal{X}_l$  represents the subset of labeled samples. Different from previous oscillatory correlation models [22], [30], here, the global inhibitor has a different dynamics. Instead of inhibiting all neurons every time a neuron pulse, only neurons representing labeled samples (neurons representing samples in  $\mathcal{X}_l$ ) can activate the global inhibitor. Moreover, if a neuron is associated to a labeled data point belonging to class 1, for instance, the global inhibitor will inhibit all other neurons except those also associated

to samples labeled as class 1. Thus, due to its conditional inhibition dynamics, the global inhibitor allows a faster local synchrony when compared to unconditional global inhibitors.

Generally speaking, the model's dynamics can be described as follows. Due to the excitatory connections modeled by Equation (5), groups of neurons (nodes) densely connected have their firing activity synchronized. At the same time, neurons representing labeled samples propagate their labels to their neighbors when synchrony is reached. On the other hand, because of the presence of the global inhibitor (Equation (7)), whenever a neuron representing a labeled sample fires, the global inhibitor generates an inhibitory signal which propagates to the whole network, thus the firing activity of neurons coding distant samples or samples from different classes are not synchronized. For this reason, the proposed model is able to perform label propagation through the network to all unlabeled samples.

An important characteristic of this approach is its simple dynamics and fast synchronization, which result in an efficient algorithm. Moreover, the dynamics of networks of I&F neurons have been widely studied [24], [39]–[44] and the parameters of these networks to ensure fast synchronization can be easily determined.

Section III presents the experiments with the proposed model using synthetic and real networks.

### III. EXPERIMENTS AND RESULTS

In this section, we present results of the introduced model. The simulations are divided into two parts. The first part consists of experiments using several synthetic data sets while the second part consists of experiments using real examples. According to the discussion in Section II, the following parameters are held constant in all experiments:  $\mu_e = \mu_y = 0.2$ ,  $I_i = 1.1$ , and  $\theta = 1.0$ . Thus, the only ad hoc parameter of our model that needs to be set according to the input data set is the threshold  $\sigma$  used to generate the graph (see Section II-A). The stopping criteria adopted for all simulations are the maximum number of iteration cycles ( $t_{max}$ ) or complete classification (when all unlabeled samples are associated to a label).

#### A. Semi-Supervised Classification on Synthetic Data

The first simulation illustrates the model's label propagation dynamics by using the data set shown by Figure 2. This synthetic data set consists of 1,000 points equally divided

into two linearly non-separable classes from which 10 samples of each class (2%) are pre-labeled. In this simulation,  $\sigma$  was set to 0.33.

Figure 3 depicts a raster plot of the 1,000 neurons representing each data set sample. The first 500 neurons (1-500) represent the class identified by red “ $\times$ ” in Figure 2, while neurons from 501 to 1,000 represent the second class shown by blue “+” in the same figure.

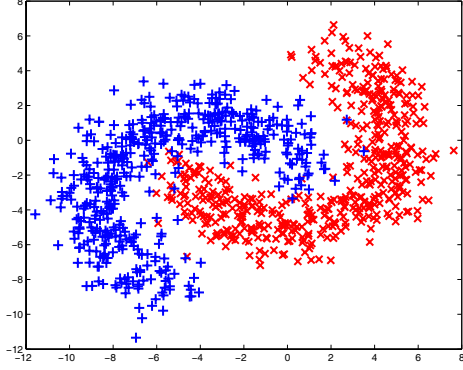


Fig. 2. Input data set illustration. This data set consists of two linearly non-separable classes with 500 samples each.

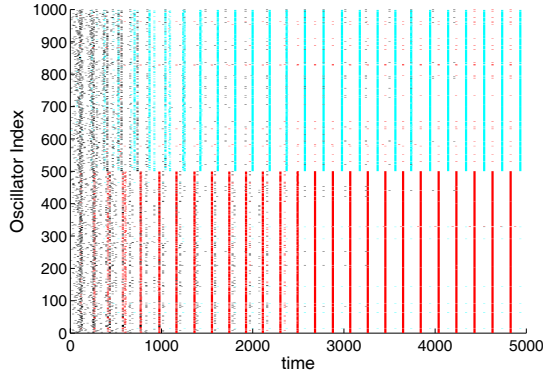


Fig. 3. Raster Plot from the simulation using the data set presented in Figure 2.

At the beginning, the neurons are not synchronized. However, modeled as an oscillatory correlation network, neighbor neurons start to synchronize with each other at the same time that distant groups of neurons remain desynchronized due to the global inhibitor.

In addition, while groups of neurons get their firing activity synchronized, those neurons representing pre-labeled samples propagate their labels to their neighbors. After a certain time, due to the synchronization/desynchronization property and the label propagation mechanism, all samples (neurons) are classified according to their pre-labeled samples.

Figures 4(a)-(f) also illustrate the label propagation process. Item (a) presents the input data set in which 2% of samples are labeled. The additional items present different instants of time, where one can observe how the labels are propagated through the data set. Figure 4(f) shows the final

result of this simulation. The final correct classification rate is approximately 97%.

To test the sensitivity of our model to different network connection densities, we perform an experiment varying the connection threshold  $\sigma$  to the same input data presented in Figure 2. We have found that the connection threshold  $\sigma$  can be computed using the information of the network average degree  $\langle k \rangle$ . It is possible to observe that a better result is achieved when  $\sigma = 0.3$ , which represents an average degree  $\langle k \rangle = 27$ . The worst case considered was found setting  $\sigma = 0.6$ , which led to a network with  $\langle k \rangle = 92$ . Figure 6 presents the results of this simulation. The same experiment was performed considering another data set and the best results were obtained when  $\sigma$  was chosen in order to generate a network with  $\langle k \rangle = 25 \pm 10$ . Thus, it is an indication that the network average degree can be used as a parameter to define an optimal, or at least, an initial value for  $\sigma$ .

In order to establish an initial value for  $\sigma$ , one can randomly select as few as 1% of the samples from the data set and calculate the distance distributions from those points. Based on this value, a  $\sigma$  can be set to ensure that, at least, a network with a certain  $\langle k \rangle$  is generated.

Figure 5 shows the temporal evolution of the model for three distinct values of  $\sigma$ . When  $\sigma = 0.1$ , the network average degree is small, thus the neighborhood of each neuron has just a few units which leads to a slow synchronization process. This phenomenon can be observed in the temporal evolution of the black line in Figure 5, where the correct classification rate increases slowly.

When  $\sigma$  is set to a larger value, ( $\sigma = 0.6$ , in Figure 5), network average degree  $\langle k \rangle$  becomes larger and the community of neurons are not easily separated. By observing the red line of Figure 5 we see that the correct classification rate oscillates near 0.5. We can explain these results as follows. Due to the large connectivity in the network, all neurons tend to synchronize with each other, which means, a global synchronization is achieved. In this situation, the label of just one class (the dominant class) is propagated to almost all neurons.

In the experiment of Figure 5, the best result is achieved when  $\sigma = 0.33$ . The best  $\sigma$  value was obtained by a further analysis (Figure 6) over the results just presented. The blue line in Figure 5 presents the temporal evolution of the correct classification rate achieved by our model when  $\sigma = 0.33$ . In this case, the synchronization among neurons is quickly obtained and the labels are correctly propagated through the network leading to a classification rate of  $\sim 97\%$ .

To conclude this section, we present a set of simulations using synthetic data sets with different data distributions as we show in Figure 7. Such data sets were built by using a Matlab toolbox named PRTTools [45] in order to generate scenarios close to real data distributions.

Data set 1 (Figure 7(a)) consists of 1,000 samples drawn from two linearly non-separable classes having 300 and 700 samples, respectively. Data set 2 (Figure 7(b)) consists of five Gaussian classes with 300, 200, 250, 300, and 150

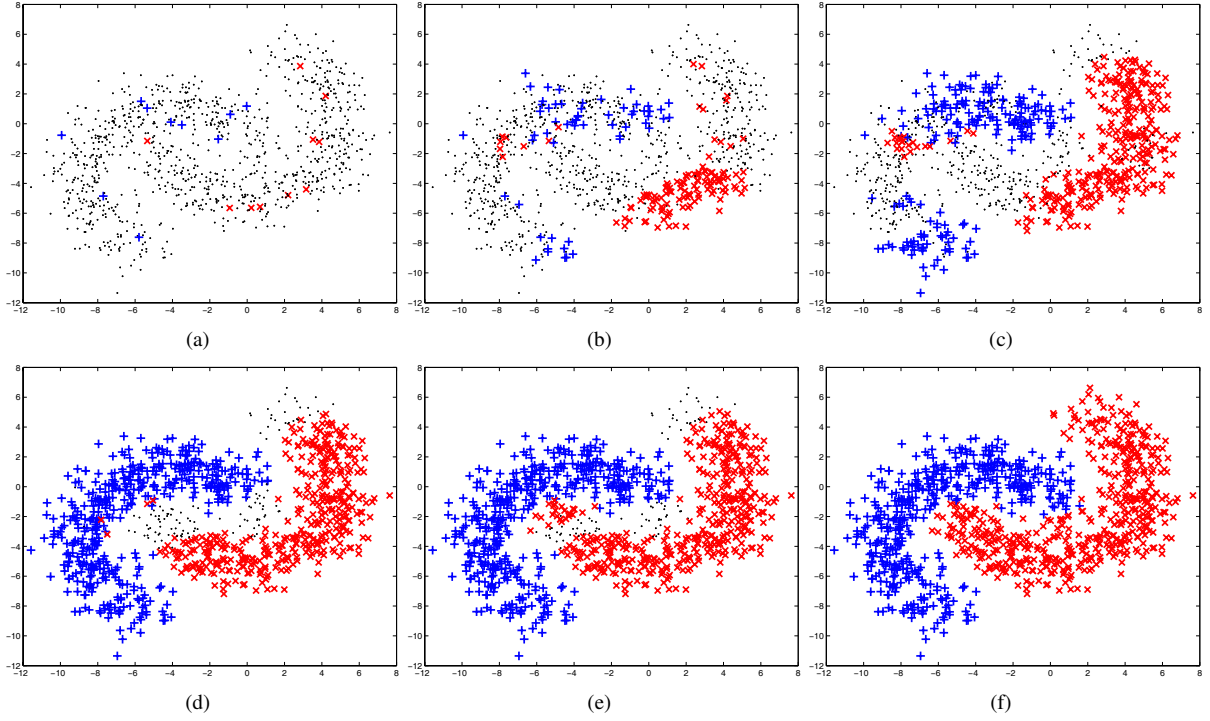


Fig. 4. Label propagation illustration. (a) The data set presented in Figure 2 where only 2% of samples are labeled (input). (b) A snapshot at  $t = 300$ . (c) A snapshot at  $t = 700$ . (d) A snapshot at  $t = 1200$ . (e) A snapshot at  $t = 2,000$ . (f) A snapshot at  $t = 5,000$  (final result – classification achieved by our model).

samples, respectively. Data set 3 (Figure 7(c)) consists of two classes of 300 samples each. Data set 4 (Figure 7(c)) consists of two classes, one with 300 samples and the other with 400 following the Highleyman distribution. Data set 5 (Figure 7(e)) comprises two classes of 800 and 400 samples, respectively. Finally, data set 6 (Figure 7(f)) consists of three classes with 400 samples each. In this data set, a class does not correspond to a cluster, thus, the cluster assumption is not presented.

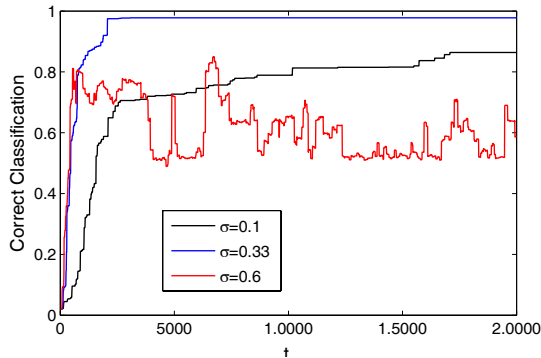


Fig. 5. Classification evolution (label propagation) for different values of  $\sigma$ . Each line represents the correct classification rate.

Figures 8(a-b) present the experimental results using the data sets depicted in Figure 7. For each simulation, the labeled data items are randomly chosen from the data set in a proportion of 1% to 10%. The remaining samples are left

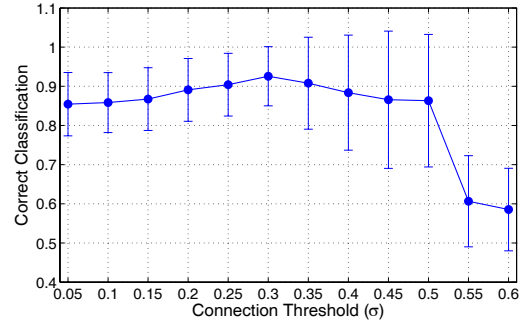


Fig. 6. Correct classification rate vs. connection threshold ( $\sigma$ ) for the data set presented in Figure 2. Each point in the trace is averaged by 200 realizations.  $t_{max} = 20,000$ . The error bars represent standard deviations.

unlabeled. We also guarantee that at least one sample of each class is labeled. We see that the model achieves good results even when only 1% of the samples are labeled. However, it is worth noting that the correct classification rate is improved when the labeled subset is increased. With just 1% of the examples pre-labeled, for data set 1, the system achieves a correct classification rate of  $\sim 85\%$ . For a more difficult data set, such as the one in Figure 7(d), the system achieves a correct classification rate of 70% with the same amount of pre-labeled data. The results presented here are the average result of 200 realizations where the model parameters were held constant and a different labeled subset was generated for each realization.



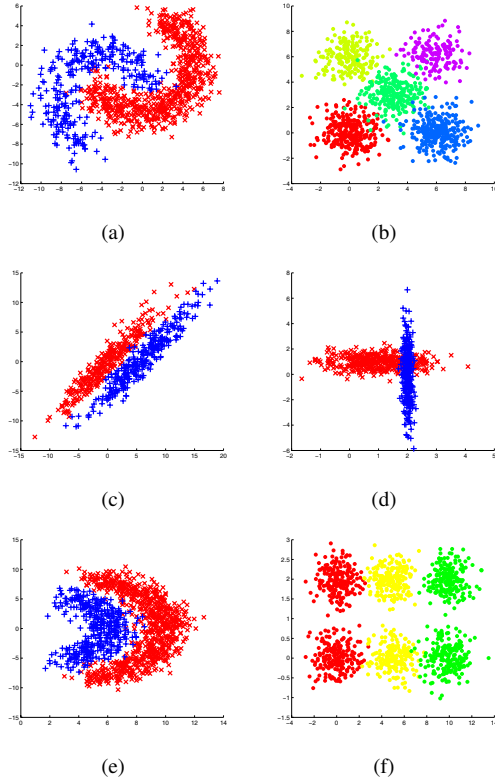


Fig. 7. Synthetic data sets used in simulations. (a) Banana-shape data set with 1,000 samples in which 300 belongs to the blue class and 700 to the red one. (b) Gaussian data set with 300, 200, 250, 300, and 150 samples in each class, respectively. (c) Two classes with 300 samples each. (d) 700 samples, 300 in the blue class and 400 in the red one. (e) 1,200 samples, 800 and 400 in the red and blue, respectively. (f) Three classes with 400 samples each.

### B. Semi-Supervised Classification on Real Data

We now turn the method's evaluation to real-world data sets. For this analysis, we select four benchmark data sets proposed and studied in [2]: USPS, COIL, BCI, and TEXT<sup>1</sup>. Table I presents a summary of these data sets.

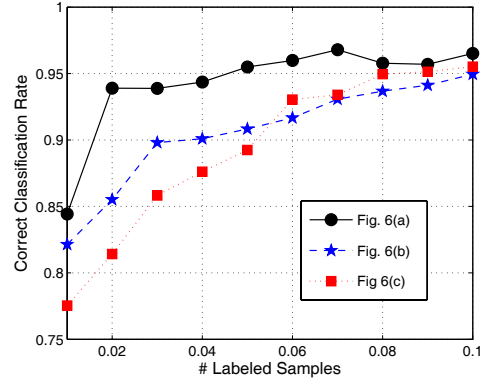
TABLE I  
REAL-WORLD DATA SET DESCRIPTION.

Dataset	Classes	Dimension	Samples
USPS	2	241	1,500
COIL	6	241	1,500
BCI	2	117	400
TEXT	2	11,960	1,500

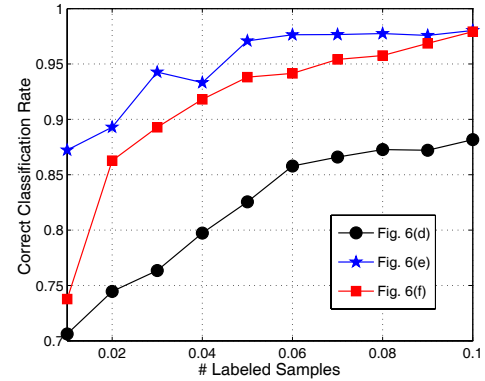
For the sake of comparison, we follow the same methodology adopted in [2] in which the labeled subset for each experiment consists either of 10 or 100 randomly selected samples.

In [2], the authors have tested and compared several techniques. Here, we compare our approach to the worst and the best results reported in [2]. It is worth noting that in [2] the authors have reported only the average results for each

<sup>1</sup>Available online at <http://www.kyb.tuebingen.mpg.de/ssl-book>



(a) Results for the data sets showed in Figure 7(a-c)



(b) Results for the data sets showed in Figure 7(d-f)

Fig. 8. Semi-supervised classification results in the data sets presented in Figure 7 with a different number of pre-labeled samples. Each point in the plot is the average of 200 realizations.  $t_{max} = 10,000$ .

technique and no information about the standard deviations was provided.

For each data set, we present the average correct classification rate for 200 realizations and the standard deviation for the two mentioned scenarios (10 and 100 labeled samples).

The USPS data set is a subset derived from the USPS handwritten database. It consists of 1,500 samples equally divided into two classes. The feature dimension of this data set is 241. The first simulation is carried out using only 10 labeled samples. With this input, our approach achieves a correct classification rate of  $80.65 \pm 0.91\%$ . When 100 labeled samples is considered, the classification rate reaches  $84.77 \pm 1.83\%$  on average. In the benchmark results reported in [2], the worst result with 10 and 100 labeled samples were 74.64% and 90.23%, respectively. The best result with 10 and 100 were 83.93% and 95.32% respectively.

The COIL data set was obtained from the Columbia object image library, which is a set of color images of several objects. It consists of 1,500 samples divided into six classes. The feature dimension of this data set is 241. For this data set, our model achieves  $36.83 \pm 7.61\%$  and  $81.49 \pm 3.43\%$  to 10 and 100 labeled samples, respectively. These results are also coherent to those presented in [2], in which the worst

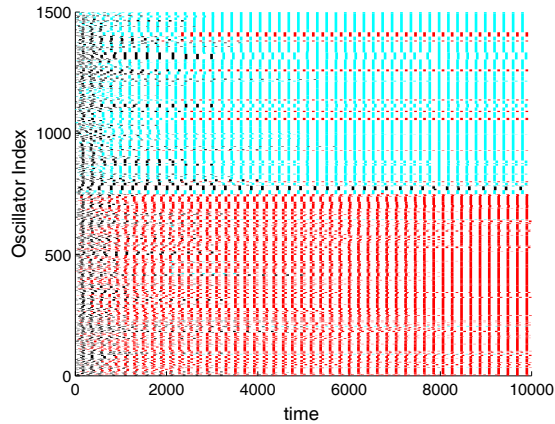


Fig. 9. Raster Plot from the simulation using the TEXT data set.

results to 10 and 100 were respectively, 32.50% and 71.29%, and the best results were 45.46% and 90.39%.

The third experiment is performed using the BCI data set. This data set was generated from the studies of a brain computer interface development. It consists of two classes of 200 samples each. The feature dimension is 117. Here, our approach achieves a correct classification rate of  $51.03 \pm 1.75\%$  and  $64.07 \pm 2.15\%$  for 10 and 100 labeled samples respectively. In [2], the worst reported results were 49.64% and 52.11%, and the best results were 52.05% and 66.75% for 10 and 100, respectively.

In the last experiment, we use the TEXT data set. This is a Newsgroup data set comprising 1,500 samples with two classes (750 samples in each class). The feature dimension of this data set is 11,960. For this data set, the best results reported in [2] were 72.85% and 76.91% to 10 and 100 labeled samples, respectively. Our model achieves  $79.77 \pm 11.82\%$  and  $94.34 \pm 2.69\%$  for 10 and 100 labeled samples, respectively. Such results show that our technique can be successfully used in real-world scenarios. In Figure 9, we also provide a raster plot depicting the synchronization and the label propagation phenomena over the considered classes.

Tables II and III summarize this section's results. In these tables one can see that our model can achieve good results when performing on real data sets. Moreover, in the TEXT data set, our model achieved the highest classification result among all techniques analyzed in [2].

TABLE II

RESULTS USING 10 LABELED SAMPLES. THIS TABLE SHOWS OUR RESULTS AVERAGED FROM 200 REALIZATIONS COMPARED TO THE WORST AND THE BEST RESULTS REPORTED IN [2].

Dataset	Our result	Worst result [2]	Best result [2]
USPS	80.65%	74.64%	83.93%
COIL	36.83%	32.50%	45.46%
BCI	51.03%	49.64%	52.05%
TEXT	79.77%	54.68%	72.85%

TABLE III

RESULTS USING 100 LABELED SAMPLES. THIS TABLE SHOWS OUR RESULTS AVERAGED FROM 200 REALIZATIONS COMPARED TO THE WORST AND THE BEST RESULTS REPORTED IN [2].

Dataset	Our result	Worst result [2]	Best result [2]
USPS	84.77%	90.23%	95.32%
COIL	81.49%	71.29%	90.39%
BCI	64.07%	52.11%	66.75%
TEXT	94.34%	67.17%	76.91%

#### IV. CONCLUDING REMARKS

In this paper, we have proposed a new graph-based method for semi-supervised learning. In contrast to previous graph-based models, which rely on a common regularizer framework, our model is based on the fast synchronization of neurons. Here, the labels from pre-labeled data samples are propagated through the graph by means of the emerging synchronization dynamics of the network.

In addition, our model has only one ad hoc parameter that needs to be set up ( $\sigma$ ). This parameter defines the connection threshold of the graph and can be chosen in order to generate a graph with a certain  $\langle k \rangle$ . From our studies, we found that our model can achieve good results when  $\langle k \rangle = 25 \pm 10$  and we also provide some insights of how to set  $\sigma$  efficiently.

Owing to neural dynamical nature, our model is an interesting approach to deal with dynamical data. For example, if more samples are added or removed from the data set, we do not need to re-execute the entire algorithm. This data change can be observed as a perturbation to the dynamical system which will reach a new equilibrium after a couple of iterations. Thus, we believe that our model can be extended as an inductive classification method. However, how to add new data points into the already generated graph is now under investigation and further research is still needed. This study is interesting to deal with real situations where new data points or even new labels are aggregated to the existing data dynamically.

As a future work, we seek to address the following open issues. First, the stopping criteria are still not completely defined. In all experiments, we adopted two criteria: maximum iteration cycles or complete classification (when all unlabeled samples are associated to a label). Although these criteria were sufficient to reach good classification results, we believe that this point deserves further analysis. Second, how to generate a proper graph to represent a given data set is not a well developed area and also needs to be investigated. Third, mislabeled samples were still not taken into account and also should be considered in order to allow the model to learn from imperfect data. Connections between pre-labeled samples could have distinct weights in order to facilitate the synchrony between different clusters representing the same class. Finally, as we have already mentioned, how to extend our model as an inductive classification method should also be investigated.

In summary, we believe that using dynamical models as a base to develop new machine learning methods is a quite

promising research field and this work has provided some efforts towards this direction.

#### ACKNOWLEDGMENTS

This research was supported by São Paulo Research Foundation (FAPESP) and by the Brazilian National Research Council (CNPq).

#### REFERENCES

- [1] X. Zhu, "Semi-supervised learning with graphs," Ph.D. dissertation, School of Computer Science, Carnegie Mellon University, 2005.
- [2] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. The MIT Press, 2006.
- [3] S. Abney, *Semisupervised Learning for Computational Linguistics*. CRC Press, 2008.
- [4] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine Learning*, vol. 39, pp. 103–134, 2000.
- [5] A. Fujino, N. Ueda, and K. Saito, "A hybrid generative/discriminative approach to semi-supervised classifier design," in *AAAI-05, Proceedings of the Twentieth National Conference on Artificial Intelligence*, 2005, pp. 764–769.
- [6] A. Demiriz, K. P. Bennett, and M. J. Embrechts, "Semi-supervised clustering using genetic algorithms," in *Proceedings of Artificial Neural Networks in Engineering (ANNIE-99)*. ASME Press, 1999, pp. 809–814.
- [7] R. Dara, S. Kremer, and D. Stacey, "Clustering unlabeled data with some improves classification of labeled real-world data," in *Proceedings of the World Congress on Computational Intelligence (WCCI)*, 2002, pp. 2237–2242.
- [8] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory - COLT-98*. ACM Press, 1998, pp. 92–100.
- [9] T. M. Mitchell, "The role of unlabeled data in supervised learning," in *Proceedings of the Sixth International Colloquium on Cognitive Science*, 1999.
- [10] Z.-H. Zhou and M. Li, "Tri-training: exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.
- [11] —, "Semisupervised regression with cotraining-style algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 11, pp. 1479–1493, Nov. 2007.
- [12] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley-Interscience, September 2008.
- [13] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the Twentieth International Conference on Machine Learning*, 2003, pp. 912–919.
- [14] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems*, vol. 16. MIT Press, 2004, pp. 321–328. [Online]. Available: <http://www.kyb.tuebingen.mpg.de/bs/people/weston/localglobal.pdf>
- [15] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," Carnegie Mellon University, Pittsburgh, Tech. Rep. CMU-CALD-02-107, 2002. [Online]. Available: <http://citeseer.ist.psu.edu/581346.html>
- [16] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55–67, Jan. 2008.
- [17] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, pp. 19–26.
- [18] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semisupervised learning on large graphs," in *Conference on Learning Theory*. Springer, 2004, pp. 624–638.
- [19] M. Belkin, N. P., and V. Sindhwani, "On manifold regularization," in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*. New Jersey: Society for Artificial Intelligence and Statistics, 2005, pp. 17–24.
- [20] T. Joachims, "Transductive learning via spectral graph partitioning," in *Proceedings of International Conference on Machine Learning*. AAAI Press, 2003, pp. 290–297.
- [21] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A universal concept in nonlinear sciences*. Cambridge University Press, 2001.
- [22] D. Terman and D. L. Wang, "Global competition and local cooperation in a network of neural oscillators," *Physica D*, vol. 81, pp. 148–176, 1995.
- [23] D. Wang and D. Terman, "Image segmentation based on oscillatory correlation," *Neural Computation*, vol. 9, pp. 805–836, 1997.
- [24] S. R. Campbell, D. L. Wang, and C. Jayaprakash, "Synchrony and desynchrony in integrate-and-fire oscillators," *Neural Computation*, vol. 11, pp. 1595–1619, 1999.
- [25] D. L. Wang and G. J. Brown, "Separation of speech from interfering sounds based on oscillatory correlation," *IEEE Transactions on Neural Networks*, vol. 10, pp. 684–697, 1999.
- [26] M. B. H. Rhouma and H. Frigui, "Self-organization of pulse-coupled oscillator with application to clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 180–195, 2001.
- [27] D. L. Wang, "Object selection based on oscillatory correlation," *Neural Networks*, vol. 12, pp. 579–592, 1999.
- [28] S. N. Wrigley and G. J. Brown, "A computational model of auditory selective attention," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1151–1163, 2004.
- [29] M. G. Quiles, D. L. Wang, L. Zhao, R. A. F. Romero, and D.-S. HUANG, "An oscillatory correlation model of object-based attention," in *The IEEE 2009 International Joint Conference on Neural Networks (IJCNN2009)*, Atlanta, Georgia, 2009, pp. 2596–2602.
- [30] D. L. Wang, "The time dimension for scene analysis," *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1401–1426, 2005.
- [31] P. Sinha, B. Balas, Y. Ostravsky, and R. Russell, "Face recognition by humans: Nineteen results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1948–1962, 2006.
- [32] R. Torres and A. Falcão, "Content-based image retrieval: Theory and applications," *Journal of Theoretical and Applied Computing (RITA)*, vol. 13, no. 2, pp. 161–185, 2006.
- [33] A. Rocha, D. Hauage, J. Wainer, and S. Goldenstein, "Automatic fruit and vegetable classification from images," *Computer and Electronics in Agriculture (COMPAG)*, vol. 70, no. 1, pp. 96–104, January 2010.
- [34] A. Rocha and S. Goldenstein, "From binary to multi-class: divide to conquer," in *Intl. Conference on Computer Vision Theory and Applications (VISAPP)*, 2009.
- [35] —, "Progressive randomization: Seeing the unseen," *Computer Vision and Image Understanding (CVIU)*, vol. 114, no. 3, pp. 349–362, March 2010.
- [36] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, 2006.
- [37] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.
- [38] D. L. Wang, "Emergent synchrony in locally coupled neural oscillators," *IEEE Transactions on Neural Networks*, vol. 6, pp. 941–948, 1995.
- [39] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645–1662, 1990.
- [40] Y. Kuramoto, "Collective synchronization of pulse-coupled oscillators and excitable units," *Physica D*, vol. 50, pp. 15–30, 1991.
- [41] A. Corral, C. J. Pérez, A. Díaz-Guilera, and A. Arenas, "Synchronization in a lattice model of pulse-coupled oscillators," *Physical Review Letters*, vol. 75, no. 20, pp. 3697–3700, 1995.
- [42] J. Hopfield and A. V. M. Herz, "Rapid local synchronization of action potentials: Toward computation with coupled integrate-and-fire oscillator neurons," *Proceedings of the National Academy of Sciences of the USA*, vol. 92, pp. 6655–6662, 1995.
- [43] A. Díaz-Guilera, C. J. Pérez, and A. Arenas, "Mechanisms of synchronization and pattern formation in a lattice of pulse-coupled oscillators," *Physical Review E*, vol. 57, no. 4, pp. 3820–3828, 1998.
- [44] M. Timme, T. Geisel, and F. Wolf, "Speed of synchronization in complex networks of neural oscillators: Analytic results based on random matrix theory," *Chaos*, vol. 16, pp. 015 108:1–15, 2006.
- [45] R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. Tax, and S. Verzaikov, "Prtools 4.1, a matlab toolbox for pattern recognition," Delft University of Technology, Tech. Rep., 2007.