# ATTRACTION FORCES BASED SEMI-SUPERVISED LEARNING

**Thiago Henrique Cupertino, Zhao Liang**
Instituto de Ciências Matemáticas e Computação - Universidade de São Paulo - São Carlos, SP
{thiagohc,zhao}@icmc.usp.br

**Abstract –** In this paper we propose a new technique for Semi-Supervised Learning based on attraction forces. The main idea behind the SSL paradigm is to perform a classification task taking into account a few labeled instances and the information provided by many unlabeled instances. Essentially, the proposed technique considers each data instance as dimensionless points on a n-dimensional space and performs their dynamics accordingly to the resultant forces. The labeled points act as fixed attraction points whereas the unlabeled ones move towards them, whereby the unlabed instances are labeled through a label propagation mechanism when they approximate a defined neighborhood region around a fixed attraction point. The technique mainly takes into account two important SSL assumptions: smoothness and cluster. The results obtained from simulations performed on artificial datasets exhibit the effectiveness of the proposed method.

**Keywords –** Semi-supervised learning, data classification, machine learning, label propagation, dynamical system and attraction force.

## 1. INTRODUCTION

The study of new machine learning techniques and applications in many and diverse areas as computer science, engineering, medical, physics, biology etc, have been increasing more and more. There are many different approaches to perform pattern recognition and classification tasks. In the branch of classification methods, a common and traditional division is the groups of *Supervised* and *Unsupervised* Learning [1]. The first, Supervised Learning (SL), aims at finding a rule that predicts the output of a given input data, i. e., it works finding relationships between input-output data pairs. On the other hand, Unsupervised Learning (UL) paradigm seeks underlying structures in a given dataset. SL works using labeled input-output pairs of data, in a way that the prediction rule is more accurate as more labeled examples is given, whereas a UL technique works with only unlabeled instances. However, a problem can occur when a technique requires labeled instances and they are hard to provide. For instance, if one wants to classify a group of web pages over the Internet accordingly to their areas of interest, e. g., news, literature, movies, sports etc, it becomes difficult to provide too much labeled examples as the Internet has billions of pages and the initial categorization of each web page must be performed by a human or expert. In this case the labeling task becomes expensive and time-consuming, a non-trivial work to perform.

To overcome this problem a different paradigm called *Semi-supervised Learning* (SSL) has been studied extensively over the past years. The main idea behind this paradigm is to perform a classification task using just a few labeled instances and the information provided by many unlabeled instances [2, 3, 4]. Hence, the SSL approach could provide higher accuracies using less human efforts and exploiting the unlabeled massive group of data.

Many SSL algorithms have been proposed [4, 3]. Some algorithms are mainly developed from generative models, including the Gaussian mixture model [5], mixture of experts [6] and extensions [7, 8], tranductive support vector machines (SVM) and semi-supervised SVM [9, 10, 11, 12, 13, 14] and boosting algorithms [15, 16, 17]. Also, the co-training is another important methodology [18]. Some techniques are graph-based as [3, 19, 20, 21]. These techniques basically map the data instances in a graph and then uses the graph underlying structure to perform and enhance the classification task. As mentioned in [4], an important problem in these algorithms is the model correctness. That is, with an incorrect model assumption, unlabeled data may even hurt the accuracy.

The SSL can be classified into two categories, namely *transductive learning* (TL) and *inductive learning* (IL). The IL seeks for a decision rule from a training set that can be applied further to new or unseen data. In contrast, TL concerns the problem of labelling the test data based on labeled data by taking both of them at the same time in the classification process. In this paper we focus on a TL algorithm.

In a SSL scenario, three important assumptions are commonly made [4]: manifold, smoothness and cluster. The manifold assumption states that the high-dimensional data lies on a low-dimensional manifold whose properties ensure more accurate density estimation and more appropriate similarity measures. The smoothness assumption states that if two points are close to each other in a high density region, then their correpondent labels should be close to each other as well. Finally, the cluster assumption states that if two points are in the same cluster, then they are likely to be of the same class (or, in other words, to have the same label). Therefore, accordingly to these two last statements, one way to correctly accomplish a SSL task is by an algorithm such that labels spread stronger in high density regions whilst, in regions between different clusters, they have weak propagation.

In this paper we propose a new technique for SSL based on attraction forces in which models the data instances as dimensionless points on a n-dimensional space and performs their motion accordingly to the resultant force applied over them. The labeled instances act as attraction points while the unlabeled instances move towards them. At a certain moment of the dynamics, the unlabeled instances receive a label propagated from the labeled points and become new attraction points. The model is very simple and, in spite of that, is effective and provided good results on the simulations performed.

This paper is organized as follows: Section 2 introduces and provides an explanation of the technique overall behavior and its mathematical modelling. Sections 3 and 4 analyze the system stability and parameters adjustment, respectively. Finally, Section 5 offers some simulation results and discussion, and Section 6 concludes the paper.

## 2. PROPOSED TECHNIQUE

A model that can fit well into the SSL smoothness and cluster assumptions is the use of *attraction forces* between labeled and unlabeled instances. The process is quite simple. The labeled instances are considered as fixed attraction points that apply attraction forces on the unlabeled instances. The latter are expected, in turn, to move towards the resultant force direction and, eventually, to converge to the closest attraction point (which provides the strongest attraction force). Once close enough (say, inside a neighborhood region $\delta$), the label propagates from the attraction point to the newest unlabeled neighbor and it becomes a new attraction point. At the end of the process is expected that all points converge to some attraction point.

One constraint for the process correct functioning is that at least one instance of each class be initially labeled. Otherwise, in the extreme case when just one class have labeled instances, all instances result having the same label. On the other hand, the advantage is that the process uses the initially labeled instances information to propagate their label to the nearest neighbors which, in turn, after being labeled, propagates it to their nearest neighbors and so on. In other words, this dynamic makes use of unlabeled data (the nearest neighbors) information to perform the classification task which is, in turn, the main idea of a SSL technique. Even more, the model allows fine adjustment as one can use any attraction force function and adjust its parameters.

Another interesting observation is that this model works with a kind of competition between labels. In some way the labeled instances compete for unlabeled ones using their attraction forces: the winner must had applied the strongest force or, in others words, the winner is the nearest instance. It can also be noticed here the important role the two previously mentioned assumptions, cluster and smoothness, play in this process.

Despite the simplicity of the model, two considerations are necessary to accomplish the above mentioned behavior and correctly classify the unlabeled instances. One of them is to guarantee that the process is stable and, the other, is to certify that the labels will propagate correctly through the unlabeled instances, in the sense that the algorithm will converge and achieve the best possible classification rates. The stability issue can be treated using similar approaches from smarm aggregation works [22, 23, 24, 25], while the label propagation dynamics can be analized in terms of the attraction force function parameters. Both are explained in the next subsections.

### *Mathematical Modelling*

Consider a dataset $D = \{L \bigcup U\}$ composed as a union between the sets of labeled $L$ and unlabeled $U$ instances, in an n-Euclidean space. The instances are modeled as points, ignoring their dimensions. We assume synchronous motion and no time delays, i. e., all points move simultaneously and know the exact positions of each other. Each unlabeled point $x_i^u$ is governed by the following equation of motion:

$$\dot{x}_i^u(t) = \sum_{j=1, j\neq i}^{|L|} f(x_i^u(t) - x_j^l(t)), i = 1, \ldots, |U|, \tag{1}$$

where f(.) is the attraction force function between labeled and unlabeled points. As defined in Eq. 1, each unlabeled instance $x_i^u$ will receive attractive forces from all labeled instances $x_j^l$ and the resultant force will be the sum of them. Therefore, the direction and magnitude of $x_i^u$ motion will be determined by all labeled instances.

In this work, we define the attraction function as a Gaussian field with parameters $\alpha$, $\beta$ and $\eta_l$:

$$f(y) = -y\frac{\alpha}{\eta_l e^{\beta\|y\|^2}}. \tag{2}$$

The attraction function has been chosen so that the more a point is close to an attractor point the more the force applied over it will be strong. Moreover, its parameters provide an easy way to adjust the function amplitude and range, which will be necessary to the correct functioning of the process as explained further in Sec. 4.

## 3. STABILITY ANALYSIS

The stability of the system as defined by Eq. 1 and 2 is analized through the Lyapunov stability method [26]. Firstly, note that the labeled instances $x^l$ are fixed points of the system, as they do no receive any attraction function and so do not move. Now, consider an unlabeled point $x_i^u$ that has been attracted in the direction of the resultant function and now is closer to a specific labeled point $x_p^l$ than to all others. Hence, $x_p^l$ is applying the strongest force over $x_i^u$ which, in turn, will putatively enter into $x_p^l$ neighborhood $\delta$ and become labeled.

Using the difference variable $e_i(t) = x_i^u(t) - x_p^l$, the Lyapunov candidate function is defined as:

$$V_i = \frac{1}{2}{e_i}^T(t)e_i(t). \tag{3}$$

Taking its derivative we have:

$$\dot{V_i} = \|e_i(t)\|\frac{e_i(t)^T}{\|e_i(t)\|}\dot{x}_i(t) = e_i(t)^T\dot{x}_i^u(t). \tag{4}$$

Substituting $\dot{x_i}$ by the expressions on Eq. 1 and 2, and dropping the time index for convenience, it results in:

$$\begin{gathered}
\dot{V_i} = -{e_i}^T\sum_{j=1,j\neq i}^{|L|}\frac{\alpha(x_i^u - x_j^l)}{\eta_l e^{\beta\|x_i^u - x_j^l\|^2}} = \\
-{e_i}^T\left[\frac{\alpha(x_i^u - x_p^l)}{\eta_l e^{\beta\|x_i^u - x_p^l\|^2}} + \sum_{j=1,j\neq i,j\neq p}^{|L|}\frac{\alpha(x_i^u - x_j^l)}{\eta_l e^{\beta\|x_i^u - x_j^l\|^2}}\right] = \\
-\frac{\alpha\|e_i\|^2}{\eta_l e^{\beta\|x_i^u - x_p^l\|^2}} + (x_p^l - x_i^u)\sum_{j=1,j\neq i,j\neq p}^{|L|}\frac{\alpha(x_i^u - x_j^l)}{\eta_l e^{\beta\|x_i^u - x_j^l\|^2}} \leq \\
-\frac{\alpha\|e_i\|^2}{\eta_l e^{\beta\|x_i^u - x_p^l\|^2}} + \left\|(x_p^l - x_i^u)\sum_{j=1,j\neq i,j\neq p}^{|L|}\frac{\alpha(x_i^u - x_j^l)}{\eta_l e^{\beta\|x_i^u - x_j^l\|^2}}\right\| = \\
-\frac{\alpha\|e_i\|^2}{\eta_l e^{\beta\|x_i^u - x_p^l\|^2}} + \|e_i\|\left\|\sum_{j=1,j\neq i,j\neq p}^{|L|}\frac{\alpha(x_i^u - x_j^l)}{\eta_l e^{\beta\|x_i^u - x_j^l\|^2}}\right\| = \\
-\alpha\|e_i\|\left(\frac{\|e_i\|}{\eta_l e^{\beta\|x_i^u - x_p^l\|^2}} - \left\|\sum_{j=1,j\neq i,j\neq p}^{|L|}\frac{(x_i^u - x_j^l)}{\eta_l e^{\beta\|x_i^u - x_j^l\|^2}}\right\|\right).
\end{gathered} \tag{5}$$

Our previously mentioned constraint states that $x_i^u$ is closer to a specific labeled point $x_p^l$ than to all others and $x_p^l$ is applying the strongest force over $x_i^u$. Therefore, we have:

$$\frac{\|e_i\|}{\eta_l e^{\beta\|x_i^u - x_p^l\|^2}} \geq \left\|\sum_{j=1,j\neq i,j\neq p}^{|L|}\frac{(x_i^u - x_j^l)}{\eta_l e^{\beta\|x_i^u - x_j^l\|^2}}\right\|, \tag{6}$$

which results:

$$\dot{V_i} < 0, \tag{7}$$

assuring that the system achieves a locally asymptotically stable equilibrium.

# 4. PARAMETERS ADJUSTMENT

After assuring the system stability, we need to assure the algorithm will converge and achieve good classification results. A way to do that is preventing the system to undergo some undesired situations. Firstly, consider the case in which a point $x_i^u$ is approximating towards a labeled attractor point $x_j^l$ and getting very close to the neighborhood limit $\delta$, where the attraction force is at at its highest amplitude before $x_i^u$ enters the neighborhood and becomes labeled. In this situation, it can occur that instead of entering the neighborhood of $x_j^l$, the point $x_i^u$ overpasses it and start *oscillating* around $x_j^l$. Other undesired situation can occur when the attraction force fields act on a big neighborhood, forcing the algorithm to *converge too fast*. In this case, a dense group of labeled points emerges too fast and attracts the majority of unlabeled points, preventing the other labels to correctly propagates as it would be done on a slower convergence. To avoid these situations, we need to correctly adjust the attraction function parameters, $\beta$, $\eta_l$ and $\alpha$. To adjust these parameters we estimate the limit values in function of the studied dataset and then set their values below those limits.

### $\beta$ *Parameter*

Firstly, the adjustment of the $\beta$ parameter is explained. Since we defined the force field shape as a Gaussian function, its maximum width can be adjusted. Consider the case in which there are two labeled points, ${x^l}_1$ and ${x^l}_2$, close to each other and be $S$ the distance between them (Fig. 1. Also, consider that there exists some unlabeled points ${x^u}_i$ between the labeled ones. As all labeled points apply the same attraction force, ${x^u}_i$ will converge to its nearest labeled point, ${x^l}_1$ or ${x^l}_2$. In this situation, all points close to ${x^l}_1$ at a distance shorter than $S/2$ is attracted towards it, while all points with at a distance longer than $S/2$ (or smaller than it considering ${x^l}_2$ as the origin) is attracted towards ${x^l}_2$. Hence, we can limit the force field of each labeled point at a half distance, $S/2$, by changing the force function tail length. To accomplish that, $S$ is set as the *shortest distance between two initially labeled instances*. Therefore, we can assure that the attraction forces will act only on a defined neighborhood estimated by the distance amplitudes of each dataset in particular.

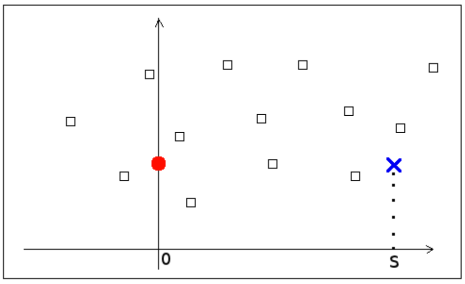


Figure 1: Two points with different labels, blue cross and red circle, at a distance $S$ from each other surrounded by unlabeled instances, squares.

The Fig. 2 depicts the attraction force function shape. Note that for higher dimensions the same function shape is extended for each coordinate. We want the point at its maximum value, $y*$, be at length $S/2$. Therefore, the information of the first derivative for each dimension is used:

$$\dot{f}(y) = -\frac{\alpha}{\eta_l}\frac{1-2\beta y^2}{e^{\beta y^2}}, \tag{8}$$

in which, after calculating $\dot{f}(y) = 0$, $y*$ is found to be:

$$y* = \frac{1}{\sqrt{2\beta}}. \tag{9}$$

With this result in hand we are able to define the attraction force function tail by doing $S/2 = 1/\sqrt{2\beta}$, or:

$$\beta = \frac{2}{S^2}. \tag{10}$$

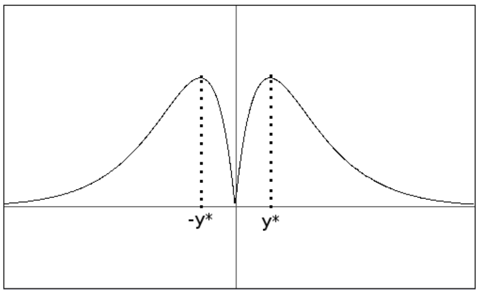


Figure 2: Attraction force function shape showing its maximum value at points y* and -y*.

### $\eta_l$ *Parameter*

Consider the situation in which the labels are already propagating and dense groups of labeled points are being formed. In this case, all points inside the labeled groups are applying attraction forces. Moreover, from $x^u$ point of view, all these forces will sum up and, consequently, the adjustment of $\alpha$ parameter will not serve anymore to limit the forces' amplitude. To avoid this situation, we need to insert a normalization parameter so that when a new point is labeled, the force applied by the labeled points of the same class is proportionally decreased. Otherwise, this can result the system oscillating because each new labeled point cause the resultant force to increase. Hence, the normalization parameter used in this work is the number of labeled points $\eta_l$ for each label $l$ at each time step. Note that the parameter value must be updated every time a point receives a new label.

### $\alpha$ *Parameter*

Now, the $\alpha$ parameter must be adjusted. For doing that, consider the extreme case in which a labeled point $x^l$ is applying the maximum force amplitude over an unlabeled point $x^u$. In this situation, $x^u$ is as close as possible to $x^l$, i. e., at the border of the labeling region $\delta$. Moreover, to guarantee covering all cases, suppose the extreme case in which only one of the $x^u$ attributes is reponsible for the distance between the two points, i. e., all the other attributes match. To avoid oscillation, it is necessary that the force applied over $x^u$ moves it inside $x^l$ neighborhood instead of throwing it around $x^l$ and, consequently, outside the neighborhood. Therefore, we want the force in this case be smaller than a limit and, if this limit is set so that $x^u$ moves towards $x^l$ no far than the actual distance between the two points, we assure that $x^u$ final position will lie inside the desired neighborhood. So, we can constraint the maximum force as:

$$y \frac{\alpha}{\eta_l e^{\beta y^2}} \leq \frac{S}{2}, \tag{11}$$

in which, setting $y = S/2$ (at the border of the $x^l$ neighborhood), $\eta_l = 1$ (when the force amplitude is at its maximum) and using $\beta = 2/S^2$ from the previous calculation, it is found that:

$$\alpha < 1.64. \tag{12}$$

***Summarized Algorithm***

In a concise form, the proposed technique can be summarized by the following algorithm:

**Algorithm 1**

```
$U \leftarrow ReadUnlabeledInstances()$
$L \leftarrow ReadInitiallyLabeledInstances()$
$AdjustParameters(\alpha, \beta, \eta_l)$
while $U \neq \emptyset$ do
  $CalculateDistancesBetweenAllPoints(U, L)$
  $CalculateAttractionForces(U, L)$
  $UpdatePointsPositions(U, L)$
  $UpdateLabels(U, L, \eta_l, \delta)$
end while
```

in which the parameters $\alpha$ and $\beta$ are adjusted accordingly the Eq. 12 and Eq. 10, respectively. $\eta_l$ is computed for each label $l$ accordingly to the simple calculation exposed in the subsection $\eta_l$ *Parameter*. The points' motion and the attraction forces are calculated using Eq. 2 and Eq. 1, respectively. The label updating funcion updates the instances labels taking into account the $\delta$ neighborhood.

***Time Complexity***

The algorithm time complexity is chiefly determined by the previously showed (Algorithm 1) computations *CalculateDistancesBetweenAllPoints* and *CalculateAttractionForces*. Both procedures have $O(n^2)$ time complexity in big O notation, in contrast to many $O(n^3)$ graph-based methods and others [3, 4]. The *while* loop must receive specific attention as it depends on the set $U$ which, in turn, depends on the algorithm convergence. As explained in Sec. 4, the parameters $\alpha$ and $\beta$ governs the convergence so as we have a trade-off between two opposite situations: a faster convergence (in detriment of accuracy) or a slower convergence (in detriment of time).

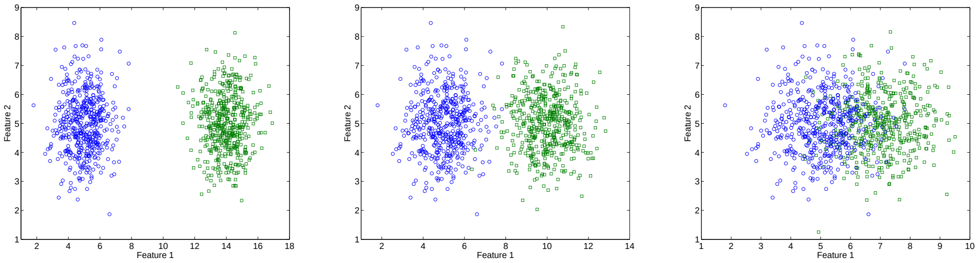
## 5. RESULTS

In this section we provide some simulation results on two kinds of artificial 2-dimensional datasets and on the benchmarks provided in Chapelle et. al [4]. In all simulations it has been used the Euclidean distance to calculate distances between pairs of points and the parameter $\delta$ has been set to $\delta = S/2$, accordingly to the analysis performed for $\beta$ in Section 4.

Firstly, we test the proposed technique on a Gaussian distribution dataset containing 1000 instances equally distributed between two classes. This dataset is depicted on Fig. 3. Three different combinations of the Gaussian set is generated by varying the means of each class (in every case it has been kept an identity covariance matrix): $(5, 5), (14, 5)$, $(5, 5), (10, 5)$ and $(5, 5), (7, 5)$, as can be seen on Figs. 3a, 3b and 3c, respectively. These combinations gradually violate the SSL assumptions as the clustering and smoothness become less defined, thus making the SSL a more difficult task. The second dataset is a two moon dataset (Fig. 4) which was generated using the PRTools toolbox [27]. The data is uniformly distributed along the moons and is superimposed with a normal distribution with standard deviation in all directions. In order to violate the SSL assumptions, it has been generated using 3 different standard deviations: 0.3, 0.6 and 1, as can be seen on Figs. 4a, 4b and 4c. This dataset also has 1000 instances equally distributed between two classes.

Tables 1 and 2 show the classification erros from the simulation results. The classification error is defined as the fraction of incorrectly labeled instances over the whole dataset. To perform the tests, it has been created 10 different splits for each set. Each of these splits contains 10 different initially labeled instances ranging from equally balanced (5 labeled instances for each classes) to completely unbalanced (9 labeled instances for one class and 1 labeled for the other). It can be seen on the tables that the technique achieved good classification results, with classification error ranging from 0 to 25% (correponding to 250 misclassified instances). It is worth noting that the technique is very sensitive to the balancing of the initially labeled examples. For instance, consider the test errors on Table 2 for set number 2 and splits $1$ ($0.1\%$), $6$ ($0.1\%$), $5$ ($2.0\%$) and $10$ ($4.7\%$). The fist two results correspond to equally balanced initial labeled instances while the last two results correspond to completely unbalanced initial labeled instances which, consequently, compose the group of the worts results for set 2 (including splits $4$ ($2.7\%$) and $9$ ($3.1\%$), which corresponds to $8$ to $2$ unbalanced labeled instances). However, all of them are competitive small error rates.

It also can be noted that the classification performance decreases, as one can expect, as there are more mixture between two different classes. In this case, the mixture creates a high difficult decision region in which an attraction point lying inside it attracts, in an extreme situation, points from both classes without distinction. Finally, it has been observed that the technique is also sensitive to the initial disposal of the labeled instances. When two initially labeled instances is too close to each other, the value of parameter $S$ becomes very small and so the range of the attraction forces. In this case, the system convergence is very slow. It should be noted that on the parameters' adjustment it has been considered the extreme limit cases which, in turn, guaratees a slow convergence but the most accurate results. Nevertheless, for specific cases one can adjust the function parameters accordingly to the dataset studied so that a very slow convergence is avoided.



(a) Set 1.

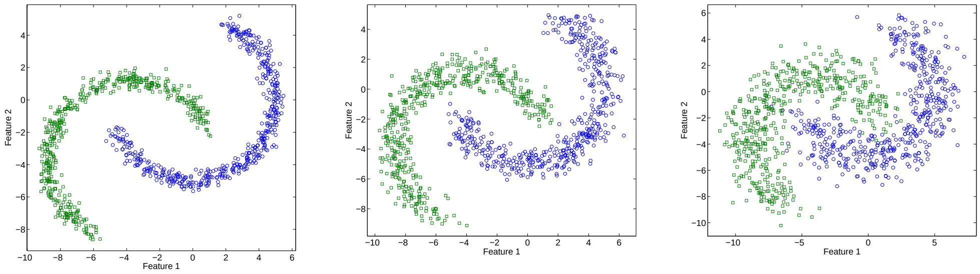
(b) Set 2.

(c) Set 3.

Figure 3: Examples of dataset for 3 artificial 2-dimensional Gaussian data.

Table 1: Gaussian set classification error rates (%).

| | Split | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Set** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | **Mean** |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 5.1 | 11.8 | 22.9 | 19.8 | 4 | 0.8 | 2.1 | 19.8 | 22.9 | 11.8 | 12.1 |
| 3 | 10.1 | 20.0 | 17.4 | 32.8 | 48.7 | 60.5 | 11.7 | 21.7 | 10.7 | 16.7 | 25.0 |



(a) Set 1.

(b) Set 2.

(c) Set 3.

Figure 4: Examples of dataset for 3 artificial 2-dimensional two moon data.

To provide comparisons for our technique we perform simulations on artificial and real-world data sets on the well-known benchmarks provided by [4]. These benchmarks provide many desired data characteristics to validade semi-supervised algorithms, such as balanced and imbalanced classes, binary and multi-classes, sparse and high-dimensional data, large sets and misleading assumptions (cluster, manifold and smoothness). The datasets range from $400$ (BCI) to $1500$ instances, from $117$ (BCI) to $241$ attributes and from $2$ to $6$ (COIL) classes. The results are showed on Table 3 for 10 and 100 initial labeled instances. It can be observed that our technique performed well specially on cases of $g241c$ and $g241d$ datasets. These are Gaussian distributed datasets which benefits the omnidirectional force fields characteristic. It is also worth to mention the good results obtained on $BCI$ dataset, a collection of human brain signals acquired by electroencephalography. On the other cases

Table 2: Two moon set classification error rates (%).

| | **Split** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Set** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | **Mean** |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.1 | 1.7 | 0.9 | 2.7 | 2.0 | 0.1 | 17.4 | 0.9 | 3.1 | 4.7 | 3.4 |
| 3 | 23.8 | 11.2 | 1.7 | 6.5 | 4.1 | 6.5 | 1.6 | 46.6 | 6.2 | 6.6 | 11.5 |

Table 3: Predictive errors (%) on datasets provided in Chapelle et. al [4].

| | 10 **initial labeled instances** | | | | | | 100 **initial labeled instances** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **g241c** | **g241d** | **Digit1** | **USPS** | **COIL** | **BCI** | **g241c** | **g241d** | **Digit1** | **USPS** | **COIL** | **BCI** |
| **1-NN** | 47.88 | 46.72 | 13.65 | 16.66 | 63.36 | 49.00 | 43.93 | 42.45 | 3.89 | 5.81 | 17.35 | 48.67 |
| **SVM** | 47.32 | 46.66 | 30.60 | 20.03 | 68.36 | 49.85 | 23.11 | 24.64 | 5.53 | 9.75 | 22.93 | 34.31 |
| **MVU + 1-NN** | 47.15 | 45.56 | 14.42 | 23.34 | 62.62 | 47.95 | 43.01 | 38.20 | 2.83 | 6.50 | 28.71 | 47.89 |
| **LEM + 1-NN** | 44.05 | 43.22 | 23.47 | 19.82 | 65.91 | 48.74 | 40.28 | 37.49 | 6.12 | 7.64 | 23.27 | 44.83 |
| **QC + CMR** | 39.96 | 46.55 | 9.80 | 13.61 | 59.63 | 50.36 | 22.05 | 28.20 | 3.15 | 6.36 | 10.03 | 46.22 |
| **Discrete Reg.** | 49.59 | 49.05 | 12.64 | 16.07 | 63.38 | 49.51 | 43.65 | 41.65 | 2.77 | 4.68 | 9.61 | 47.67 |
| **TSVM** | 24.71 | 50.08 | 17.77 | 25.20 | 67.50 | 49.15 | 18.46 | 22.42 | 6.15 | 9.77 | 25.80 | 33.25 |
| **SGT** | 22.76 | 18.64 | 8.92 | 25.36 | n/a | 49.59 | 17.41 | 9.11 | 2.61 | 6.80 | n/a | 45.03 |
| **Cluster-Kernel** | 48.28 | 42.05 | 18.73 | 19.41 | 67.32 | 48.31 | 13.49 | 4.95 | 3.79 | 9.68 | 21.99 | 35.17 |
| **Data-Dep. Reg.** | 41.25 | 45.89 | 12.49 | 17.96 | 63.65 | 50.21 | 20.31 | 32.82 | 2.44 | 5.10 | 11.46 | 47.47 |
| **LDS** | 28.85 | 50.63 | 15.63 | 17.57 | 61.90 | 49.27 | 18.04 | 23.74 | 3.46 | 4.96 | 13.72 | 43.97 |
| **Laplacian RLS** | 43.95 | 45.68 | 5.44 | 18.99 | 54.54 | 48.97 | 24.36 | 26.46 | 2.92 | 4.68 | 11.92 | 31.36 |
| **CHM (normed)** | 39.03 | 43.01 | 14.86 | 20.53 | n/a | 46.90 | 24.82 | 25.67 | 3.79 | 7.65 | n/a | 36.03 |
| **LGC** | 45.82 | 44.09 | 9.89 | 9.03 | 63.45 | 47.09 | 41.64 | 40.08 | 2.72 | 3.68 | 45.55 | 43.50 |
| **Proposed Method** | 41.78 | 40.63 | 36.44 | 32.03 | 72.95 | 31.41 | 21.89 | 21.30 | 15.42 | 19.51 | 49.83 | 46.85 |

our algorithm experiences not so good results. This can be due to the instances distribution which difficults the attraction force approach.

## 6. CONCLUSIONS

This work presented a new semi-supervised learning technique based on attraction forces between data instances. Firstly, a matemathical model has been formulated to verify the system stability through Lyapunov method. Subsequently, some heuristics have been used to adjust the attraction function parameters based on the system desired dynamical behavior. Finally, simulation results have been exposed and discussed. It has been verified that the technique is sensitive to the initially labeled instances balancing and placement. However, the proposed technique showed good classification results even when SSL smoothness and cluster assumptions were not completely satisfied. As future work it will be interesting to study the behavior of the system when different distance functions are taken into account. In this work it has been used the Euclidean distance although it is known that this measure has problems in higher dimensional spaces. Also, it could be explored different heuristics and refinements for the parameters adjustment and the definition of the attraction point neighborhood, e. g. taking into account the data distribution on that region, inside which labels propagate. Lastly, the balance between the relatively technique simplicity and the good classification rates, shows that the proposed method can be well suited for performing accurate SSL tasks.

## REFERENCES

[1] T. M. Mitchell. *Machine learning.*. McGraw Hill series in computer science, first edition, 1997.

[2] M. Seeger. "Learning with Labeled and Unlabeled Data". Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, 2002.

[3] X. Zhu. "Semi-Supervised Learning Literature Survey". Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2007.

[4] O. Chapelle, B. Schölkopf and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[5] B. Shahshahani and D. Landgrebe. "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon". *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 32, no. 5, pp. 1087 –1095, sep 1994.

[6] D. J. Miller and H. S. Uyar. “A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data”. In *Neural Information Processing Systems*, pp. 571–577, 1996.

[7] K. Nigam, A. K. McCallum, S. Thrun and T. M. Mitchell. “Text Classification from Labeled and Unlabeled Documents using EM”. *Machine Learning*, pp. 1–34, 2000.

[8] A. Fujino, N. Ueda and K. Saito. “A Hybrid Generative/Discriminative Approach to Semi-Supervised Classifier Design”. In *National Conference on Artificial Intelligence*, pp. 764–769, 2005.

[9] O. Chapelle and A. Zien. “Semi-Supervised Classification by Low Density Separation”. In *Proc. 10th Int. Workshop Artif. Intell. Statist.*, pp. 57–64, 2005.

[10] R. Collobert, F. H. Sinz, J. Weston and L. Bottou. “Large Scale Transductive SVMs”. *Journal of Machine Learning Research*, vol. 7, pp. 1687–1712, 2006.

[11] T. Joachims. “Transductive Learning via Spectral Graph Partitioning”. In *International Conference on Machine Learning*, pp. 290–297, 2003.

[12] L. Xu. “Unsupervised and semi-supervised multi-class support vector machines”. In *AAAI-05, The Twentieth National Conference on Artificial Intelligence*, pp. 904–910, 2005.

[13] Z. Xu, R. Jin, J. Zhu, I. King, M. Lyu and Z. Yang. “Adaptive Regularization for Transductive Support Vector Machine”. In *Advances in Neural Information Processing Systems*, edited by Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams and A. Culotta, volume 22, pp. 2125–2133. Eds. Cambridge, MA: MIT Press, 2009.

[14] Z. Xu, R. Jin, J. Zhu, I. King and M. R. Lyu. “Efficient Convex Relaxation for Transductive Support Vector Machine”. In *Neural Information Processing Systems*, edited by J. Platt, D. Koller, Y. Singer and S. Roweis, volume 20, pp. 1641–1648. Eds. Cambridge, MA: MIT Press, 2007.

[15] N. Loeff, D. A. Forsyth and D. Ramachandran. “ManifoldBoost: stagewise function approximation for fully-, semi- and un-supervised learning”. In *International Conference on Machine Learning*, pp. 600–607, 2008.

[16] P. K. Mallapragada, R. Jin, A. K. Jain and Y. Liu. “SemiBoost: Boosting for Semi-Supervised Learning”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 2000–2014, 2009.

[17] K. Chen and S. Wang. “Semi-Supervised Learning via Regularized Boosting Working on Multiple Semi-Supervised Assumptions”. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 1, pp. 129–143, 2011.

[18] A. Blum and T. Mitchell. “Combining labeled and unlabeled data with co-training”. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT' 98, pp. 92–100, New York, NY, USA, 1998. ACM.

[19] 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 16*, pp. 321–328. MIT Press, 2004.

[20] A. Blum and S. Chawla. “Learning from Labeled and Unlabeled Data using Graph Mincuts”. In *International Conference on Machine Learning*, pp. 19–26, 2001.

[21] M. Belkin, I. Matveeva and P. Niyogi. “Regularization and Semi-supervised Learning on Large Graphs”. In *Computational Learning Theory*, pp. 624–638, 2004.

[22] V. Gazi and K. M. Passino. “Stability analysis of swarms”. *IEEE Transactions on Automatic Control*, vol. 48, pp. 692–697, 2003.

[23] K. Jin, P. Liang and G. Beni. “Stability of Synchronized Distributed Control of Discrete Swarm Structures”. In *International Conference on Robotics and Automation*, pp. 1033–1038, 1994.

[24] G. Beni and P. Liang. “Pattern reconfiguration in swarms-convergence of a distributed asynchronous and bounded iterative algorithm”. *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 485–490, 1996.

[25] Y. Liu, K. M. Passino and M. Polycarpou. “Stability analysis of one-dimensional asynchronous swarms”. *IEEE Transactions on Automatic Control*, vol. 48, pp. 1848–1854, 2003.

[26] K. M. Hangos, J. Bokor and G. Szederkényi. *Analysis and Control of Nonlinear Process Systems*. Springer, first edition, 2004.

[27] R. P. W. Duin. “PRTools Version 3.0: A Matlab Toolbox for Pattern Recognition”. In *Proc. of SPIE*, p. 1331, 2000.