Combined Active and Semi-Supervised Learning using Particle Walking Temporal Dynamics

Fabricio Breve
Institute of Geosciences and Exact Sciences (IGCE)
São Paulo State University (UNESP)
Rio Claro, Brazil
Email: fabricio@rc.unesp.br

***Abstract*—Both Semi-Supervised Leaning and Active Learning are techniques used when unlabeled data is abundant, but the process of labeling them is expensive and/or time consuming. In this paper, those two machine learning techniques are combined into a single nature-inspired method. It features particles walking on a network built from the data set, using a unique random-greedy rule to select neighbors to visit. The particles, which have both competitive and cooperative behavior, are created on the network as the result of label queries. They may be created as the algorithm executes and only nodes affected by the new particles have to be updated. Therefore, it saves execution time compared to traditional active learning frameworks, in which the learning algorithm has to be executed several times. The data items to be queried are select based on information extracted from the nodes and particles temporal dynamics. Two different rules for queries are explored in this paper, one of them is based on *querying by uncertainty* approaches and the other is based on data and labeled nodes distribution. Each of them may perform better than the other according to some data sets peculiarities. Experimental results on some real-world data sets are provided, and the proposed method outperforms the semi-supervised learning method, from which it is derived, in all of them.**

I. Introduction

Semi-Supervised Learning is a class of machine learning techniques which focus on problems where there are lots of easily acquired unlabeled data, but the process of labeling them is often expensive, time consuming, and/or requires the work of human specialists [1]–[3]. It is pathway between Supervised Learning and Unsupervised Learning. Supervised Learning methods use only labeled data in their training process, no information is taken from unlabeled data. On the other hand, Unsupervised Learning methods use all data items but any label information is wasted. Semi-Supervised Learning methods are able to take advantage of both labeled and unlabeled data in order to build better classifiers.

Active Learning is also a form of machine learning. The methods in this class are able to interactively query an human specialist (or any other kind of information source) in order to obtain the labels of selected data points. The key idea behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns [4], [5]. Active learning is also useful on problems where unlabeled data is abundant, but labeling them is difficult.

Therefore, semi-supervised learning and active learning methods share this predilection of making the most of unlabeled data. But there are also some differences. While semi-supervised learning methods exploit what the learners think they know about the unlabeled data, active learning methods attempt to explore unknown aspects [5]. For instance, semi-supervised learning methods often propagate labels through similar data points, or label the data which it has more confidence, and then retrain the algorithm with those new labeled data. On the other hand, active learning methods may query labels of the nodes they have least confidence, and then retrain the algorithm with those new labeled nodes as well. Another example: *co-training* semi-supervised methods relies on committees agreements, while *query by committee* active learning methods rely on committees disagreements [4].

Active Learning algorithms are usually split in categories according to how they choose which data points should have their labels queried. Some of the most common approaches includes: *uncertainty sampling*, in which the algorithms query the labels of data points in which they have less confidence [6]–[9]; *querying by committee*, category in which a variety of models are trained on the current labeled data, and then query for the labels of the data points in which they disagreed the most [10]–[12]; *expected model change*, in which algorithms query the labels of the instances that would lead to the greatest changes in the current model [13]; *expected error reduction*, in which algorithms query data points that lead to the largest decrease in the expected error [14], [15]; *expected output variance reduction*, which consists in labeling those points that would minimize output variance [16]; and *density-weighted methods*, in which the data density information is used to choose the data points to be queried [13], [17].

Semi-supervised learning methods are also usually arranged in categories, like generative models [18], [19], cluster-and-label techniques [20], [21], co-training and tri-training techniques [22]–[25], low-density separation models [26], and graph-based methods, which is the most active category in the recent years. It includes methods like Mincut [27], Local and Global Consistency [28], some label propagation techniques [29], [30], Particle Competition and Cooperation [31], among others.

In the Particle Competition and Cooperation method, a network is built from the data set using the Euclidean distance among data items. Particles then walk in the network trying to dominate its nodes. Particles are organized in teams. Each team represents a problem class. The competition takes place among particles of different teams, while particles from the same team cooperate with each other. Labels are spread as particles move from node to node. Particles have both exploratory and defensive behavior, and they alternate between them. Exploratory behavior is used to propagate their labels to unlabeled nodes, while the defensive behavior is used to keeps particles close to nodes already dominated by their team. The particles start from the labeled nodes, which are pre-defined according to the pre-labeled data points. There is no retraining process. Particles quickly dominate the unlabeled nodes which are closer to their respective labeled node, but there is usually a tough dispute for the nodes on frontier regions, or even dense regions without labeled nodes. Thus, classification performance could increase if the algorithm was able to query the labels of specific nodes in those regions.

In this paper, the particle competition and cooperation method [31] is extended to combine both semi-supervised and active learning features. The new method may use as few as a single pre-labeled node per class to start. A particle is generated for each labeled node. Then, the algorithm dynamically chooses nodes to have their labels queried. New particles are generated on the fly as new labels are provided, so the algorithm does not have to be restarted. Therefore, it is usually much faster than methods that require retraining after new queries. Moreover, only nodes affected by new particles will be changed, so the network quickly reaches a new equilibrium state after new queries.

Two different rules for querying the chosen unlabeled nodes are explored in this paper. The first rule is inspired on *querying by uncertainty* approaches. It uses temporal nodes domination information in order to select nodes that were the target of most disputes over time. The second rule is based on nodes distribution in the network. It uses network data distribution information, dynamically collected by the nodes as they walk, in order to select nodes that are far away from any labeled node. Those nodes are more susceptible to wrong label propagation. A single labeled node among them may avoid this problem. In problems with overlapping classes or outliers, it may be better to concentrate on dense regions without labeled nodes instead of the border nodes. The simulations reveal that each rule may perform better than the other according to some data sets peculiarities.

The remaining of this paper is organized as follows. The proposed model is described in section II. In Section III, computer simulation results are presented. Finally, some conclusions are drawn on Section IV.

II. Model Description

The proposed model uses particle competition and cooperation in undirected and non-weighted networks, which are built from the data set. A particle is generated for each labeled data item. Particles with the same label form a group, called *team*. Teammates act cooperatively to dominate nodes of the network. Different teams compete against each other to prevent rivals from entering the nodes they have dominated. Particles jump from node to node in order to increase their domination on them, but they are expelled when the selected node is dominated by another team. Particles may lose or gain strength as they walk, depending on the domination levels of the visited node.

The network is built from the data set $\chi = \{x_1, x_2, \dots, x_l, x_{l+1}, \dots, x_n\} \subset \mathbb{R}^m$, with the corresponding label set $L = \{1, 2, \dots, c\}$. The first l points $x_i (i \leq l)$ are labeled, $y_i \in L$. The remaining points $x_u (l < u \leq n)$ are left unlabeled, $y_u = \emptyset$. An undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is created, in which $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$ is the set of nodes, and $\mathbf{E}$ is the set of edges (v_i, v_j). Each node v_i corresponds to a data point x_i. Two nodes v_i and v_j are connected if v_j is among the k-nearest neighbors of v_i, or vice-versa. The Euclidean distance is used. Otherwise, v_i and v_j are disconnected. For each network node $v_i \in \{v_1, v_2, \dots, v_l\}$, corresponding to a labeled data point $x_i \in \{x_1, x_2, \dots, x_l\}$, there is a particle $\rho_i \in \{\rho_1, \rho_2, \dots, \rho_l\}$ which initial position is v_i. The algorithm needs at least one labeled node per class to start. Then, as the algorithm queries for node labels, new particles are generated for the new labeled nodes.

Each particle ρ_j has two variables. The first one is $\rho_j^\omega(t) \in [0, 1]$. It holds the particle strength, which defines how much the particle can change a node level at time t. The second variable is a distance table $\rho_\mathbf{j}^\mathbf{d}(\mathbf{t}) = \{\rho_i^{d_1}(t), \rho_i^{d_2}(t), \dots, \rho_i^{d_n}(t)\}$. Each element $\rho_j^{d_i}(t) \in [0 \quad n-1]$ holds the distance dynamically measured between the particle corresponding node and node v_i. Each particle ρ_j is created with its initial position set to its corresponding labeled node and its initial strength set to maximum, $\rho_j^\omega(0) = 1$. Particles start knowing only the distance to their corresponding labeled nodes, which is set to zero ($\rho_j^{d_i} = 0$). Other distances are set to the largest possible value ($\rho_j^{d_i} = n - 1$).

Each node v_i has a domination table $\mathbf{v}_\mathbf{i}^\omega(\mathbf{t}) = \{v_i^{\omega_1}(t), v_i^{\omega_2}(t), \dots, v_i^{\omega_c}(t)\}$. Each element $v_i^{\omega_\ell}(t) \in [0, 1]$ corresponds to the domination level from team ℓ over node v_i. The sum of the domination levels in each node is always constant, $\sum_{\ell=1}^{c} v_i^{\omega_\ell} = 1$. Domination levels are fixed for labeled nodes and variable for unlabeled nodes. Labeled nodes are always fully dominated by their corresponding team. Unlabeled nodes have their domination levels changed as particles select them to visit. Therefore, for each node v_i, the domination table $\mathbf{v}_\mathbf{i}^\omega$ is initially set as follows:

$$v_i^{\omega_\ell}(0) = \begin{cases} 1 & \text{if } y_i = \ell \\ 0 & \text{if } y_i \neq \ell \text{ and } y_i \in L \\ \frac{1}{c} & \text{if } y_i = \emptyset \end{cases} . \quad (1)$$

Each node v_i also has a *long term domination table* $\mathbf{v}_\mathbf{i}^\lambda(\mathbf{t}) = \{v_i^{\lambda_1}(t), v_i^{\lambda_2}(t), \dots, v_i^{\lambda_c}(t)\}$. Each element $v_i^{\lambda_\ell}(t) \in [0 \quad \infty]$ corresponds to long term domination level by team ℓ over node v_i. Long term domination levels can vary from zero to infinity and they never decrease. Therefore, all long term domination levels $\mathbf{v}_\mathbf{i}^{\lambda_\ell}(\mathbf{0})$ have their initial values set to zero, for all the classes ℓ no matter if the corresponding data item is labeled or unlabeled [32]. After each query, long term domination levels are reset.

At each iteration, each particle ρ_j chooses a node v_i from the neighbors of its current node. It will try to visit the selected node, which will be called *target node* from now on. There are two rules to select the target node. The *random rule* and the *greedy rule*. At each iteration, the particle randomly selects one of them with equal probabilities. Then, if the *random rule* is selected, the probabilities of the particle ρ_j choosing a neighbor v_i are defined equally to each neighbor:

$$p(v_i|\rho_j) = \frac{W_{qi}}{\sum_{\mu=1}^{n} W_{q\mu}}, \tag{2}$$

where q is the index of the current node of particle ρ_j, so $W_{qi} = 1$ if there is an edge between the current node and any node v_i, and $W_{qi} = 0$ otherwise. On the other hand, if the *greedy rule* is selected. The probabilities of the particle ρ_j choosing a neighbor v_i are defined according to the particle team domination on it, $\rho_j^{\omega_\ell}$, and the inverse of its distance, $\rho_j^{d_i}$, to the particle initial position, v_j, as follows:

$$p(v_i|\rho_j) = \frac{W_{qi} v_i^{\omega_\ell} (1 + \rho_j^{d_i})^{-2}}{\sum_{\mu=1}^{n} W_{q\mu} v_\mu^{\omega_\ell} (1 + \rho_j^{d_\mu})^{-2}}, \tag{3}$$

where q is the index of the current node of particle ρ_j and $\ell = \rho_j^f$, where ρ_j^f is the class label of particle ρ_j. Notice that the particle only stays at the target node if its team domination level is higher than those from all other teams; otherwise, a shock happens and the particle is expelled, going back to its previous location until the next iteration.

Each unlabeled node v_i selected as a target node has its domination table updated as follows:

$$v_i^{\omega_\ell}(t+1) = \begin{cases} \max\{0, v_i^{\omega_\ell}(t) - \frac{0.1\rho_j^{\omega}(t)}{c-1}\} \\ \quad \text{if } \ell \neq \rho_j^f \\ v_i^{\omega_\ell}(t) + \sum_{r\neq\ell} v_i^{\omega_r}(t) - v_i^{\omega_r}(t+1) \\ \quad \text{if } \ell = \rho_j^f \end{cases}, \tag{4}$$

where ρ_j^f represents the class label of particle ρ_j. Each particle ρ_j changes the target node v_i by increasing the domination level of its team ($v_i^{\omega_\ell}$, $\ell = \rho_j^f$) while decreasing the domination levels of other teams ($v_i^{\omega_\ell}$, $\ell \neq \rho_j^f$)). Notice that (4) does not apply to labeled nodes.

Long term domination levels $v_i^{\lambda_\ell}(t)$ are updated only when the *random rule* is selected, as follows:

$$v_i^{\lambda_\ell}(t+1) = v_i^{\lambda_\ell}(t) + \rho_j^{\omega}(t) \tag{5}$$

where ℓ is the class label of particle ρ_j.

Then, particle strength is updated at each iteration, as follows: $\rho_j^{\omega}(t) = v_i^{\omega_\ell}(t)$, where v_i is the target node, and $\ell = \rho_j^f$, i.e., ℓ is the class label of particle ρ_j.

Distance calculation is a dynamical process, so each particle ρ_j updates its distance table $\rho_j^{d_k}(t)$ at each iteration t as follows:

$$\rho_j^{d_k}(t+1) = \begin{cases} \rho_j^{d_i}(t) + 1 & \text{if } \rho_j^{d_i}(t) + 1 < \rho_j^{d_k}(t) \\ \rho_j^{d_k}(t) & \text{otherwise} \end{cases}, \tag{6}$$

where $\rho_j^{d_i}(t)$ is the distance from the current node to the particle initial node, and $\rho_j^{d_k}(t)$ is the distance from the target node to the particle initial node.

As new labels are queried, (1) is reapplied to the new labeled nodes. After the last iteration of the algorithm, each unlabeled node is labeled after the team which has the highest domination level on it, i.e., $y_i = \arg\max_\ell v_i^{\omega_\ell}(t)$.

The average maximum domination level of the nodes ($\langle v_i^{\omega\ell} \rangle$, $\ell = \arg\max_q v_i^{\omega_q}$) is used to identify when the algorithm reaches stability. This value does not converge, as there is always some dispute in the nodes in classes frontiers. But we may set a stop criterion for when there is no increase for a fair amount of iterations, usually $\approx \frac{100.n}{l}$, where n is the network size and l is the current amount of particles in the network. When the system reaches stability, it is time to query for another node label, generating a new particle to the new labeled node. This procedure is repeated until the specified amount of labeled nodes is reached. Then, the same criteria may be used to finish the algorithm, extracting the final labels.

There are two different rules to choose which node will have its label queried. Rule A uses temporal node domination information to select the unlabeled node which had more dispute over time, which may be understand as the node the algorithm has less confidence on the label it is currently assigning. Rule B chooses the unlabeled node which is currently more far away from any labeled node, according to particles dynamic distance tables.

In order to select the network node which was most disputed over time at any given time (Rule A), each node dispute level (or uncertainty) is calculated as follows:

$$u_i(t) = \frac{v_i^{\lambda_{\ell**}}(t)}{v_i^{\lambda_{\ell*}}(t)}, \tag{7}$$

where $v_i^{\lambda_{\ell*}}(t) = \arg\max_\ell v_i^{\lambda_\ell}(t)$, $v_i^{\ell**}(t) = \arg\max_{\ell,\ell\neq v_i^{\ell*}(t)} v_i^{\lambda_\ell}(t)$, and $u_i \in [0 \quad 1]$, where $u_i = 0$ means completely confidence in the label given to a node, while $u_i = 1$ means the node label is completely undefined among two or more classes. Then, the unlabeled node which was most disputed over time (most uncertain label) is defined using:

$$q(t) = \arg\max_{i, y_i = \emptyset} u_i(t). \tag{8}$$

Notice that all *long term domination tables* are redefined after each query, otherwise older disputes that no longer take place would influence the next query.

In order to select the node which is more far away from any labeled node (Rule B), the dynamic distance from each node to its closest labeled node, as taken by the particles, is calculated:

$$s_i(t) = \min_j \rho_j^{d_i}(t). \tag{9}$$

Then, the unlabeled node which is more far away from any labeled node is defined using:

$$q(t) = \arg\max_{i, y_i = \emptyset} s_i(t). \tag{10}$$

III. Computer Simulations

In order to show the effectiveness of the proposed method, some simulations are executed using some real-world data sets. In each simulation, the correct classification rate obtained by the original particle competition and cooperation method (PCC) [31] is compared with those achieved by the proposed method using both rules individually (ASL-PCC A and ASL-PCC B). The following parameters were fixed for the PCC method: $p_{grd} = 0.5$ and $\Delta_v = 0.1$. For the three methods,

$k = 5$ is fixed, i.e., each node is connected to its 5 nearest neighbors. These are optimal or near optimal parameter values for most data sets. They were obtained by empirical optimization using the grid method.

The PCC method is executed using 1% to 10% data items randomly chosen to be pre-labeled, as it requires. On the other hand, for both versions of ASL-PCC, we randomly choose only one data item per class to be pre-labeled. The remaining labeled samples are obtained through queries made each time the algorithm reaches stability. This is repeated until the defined amount of labeled items (1% to 10%) is reached. Notice that for Iris and Wine data sets (Figures 1a and 1b), simulations with only 1% and 2% labeled nodes are skipped, as only one labeled node per class would already pass the 2% mark, leaving no room for queries. Each point in the graphics from Figures 1a to 2d is the average of 100 executions with different pre-labeled nodes.

Figures 1a to 2d shows the classification performance comparison when the methods are applied to 8 different data sets. In all cases, the proposed method works better than the original PCC, with at least one rule. Rule A is the best on half of the cases and Rule B is the best on the other half. Moreover, in half of the cases, the proposed method is better no matter which rule is selected, while in the other half it requires the selection of the more adequate rule.

Rule A works better in data sets where the classes are well separated and there are not many outliers, because in these cases the dense regions are easily classified and the uncertainty is mostly in frontier regions. Rule B, on the other hand, usually works better when the classes are not well defined and/or there are outliers, because in these cases the labeled nodes are sparsely distributed.

IV. Conclusions

In this paper, semi-supervised learning and active learning features are combined into a single approach, thus a new classification method is proposed. It is inspired on the collective behavior of social animals, which protect their territories against intruding groups. Networks are built from the data sets and particles, representing labeled nodes, walk from node to node choosing the next node through an unique random-greedy rule. Particles compete against particles from other groups in order to dominate network nodes, expanding their territory and preventing invasion from other particles. On the other hand, particles from the same group act cooperatively and share their territory.

Most active learning methods requires an explicit retraining process in order to incorporate new label information, resulting from queries. The proposed method does not require retraining. New particles are created on the fly as unlabeled nodes become labeled nodes. The algorithm naturally adapts itself to the new situation. Only nodes affected by the new particles are updated, and an equilibrium state is quickly achieved again, saving execution time.

Two different rules for selecting unlabeled nodes to be queried are explored in this paper. The first one is inspired on *querying by uncertainty* approaches. Temporal nodes domination information is used to select nodes that were the target of most disputes over time. This rule is better to define frontiers in problems where classes are fairly well separated and there are not many outliers. The second rule is based on the nodes distribution in the network. Data distribution information is dynamically collected by the nodes as they walk. Nodes are selected based on how far they are from any labeled node, since those nodes are more susceptible to wrong label propagation. This approach works better in data sets where classes have some highly overlapped regions and/or many outliers. It addresses the problem of querying outliers, commonly found on *querying by uncertainty* approaches [15], [27].

Some real-world data sets are used in order to measure the performance of the proposed method. In all cases, it performs better than the semi-supervised learning approach from which it was inspired, at least when the correct querying rule is selected. Unfortunately, when the data set is unknown, there may be no clue on which querying rule would work better. In future works, this problem may be addressed through the analysis of the network structure together with the labeled nodes position in the network. Moreover, the querying rule could be switched on the fly based on temporal information, such as new queried labels and new labeled nodes distribution.

Acknowledgment

The author would like to thank the São Paulo State Research Foundation (FAPESP) and the Foundation for the Development of Unesp (Fundunesp) for the financial support.

References

[1] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005.

[2] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA: The MIT Press, 2006.

[3] S. Abney, *Semisupervised Learning for Computational Linguistics*. CRC Press, 2008.

[4] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[5] F. Olsson, "A literature survey of active machine learning in the context of natural language processing," Swedish Institute of Computer Science, Box 1263, SE-164 29 Kista, Sweden, Tech. Rep. T2009:06, April 2009.

[6] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '94. New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 3–12.

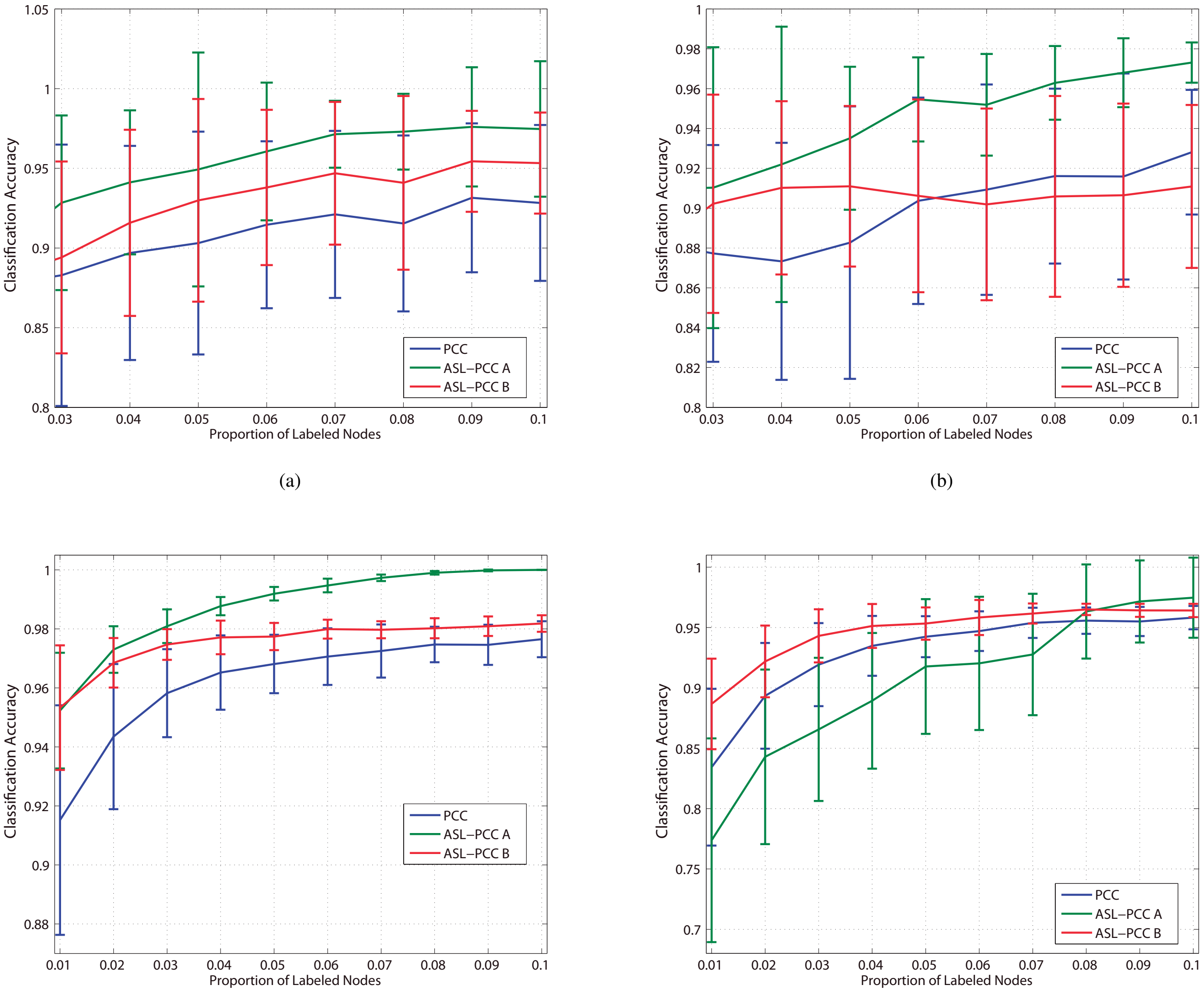
[7] D. Cohn, R. Ladner, and A. Waibel, "Improving generalization with active learning," in *Machine Learning*, 1994, pp. 201–221.

[8] G. Schohn and D. Cohn, "Less is more: Active learning with support vector machines," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 839–846.

[9] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, ser. IDA '01. London, UK, UK: Springer-Verlag, 2001, pp. 309–318.

[10] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 287–294.

[11] R. Liere and P. Tadepalli, "Active learning with committees for text categorization," in *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence*, ser. AAAI'97/IAAI'97. AAAI Press, 1997, pp. 591–596.



(a)

(b)

(c)

(d)

Fig. 1. Correct classification rate and standard deviation comparison when the methods are applied varying the amount of labeled nodes in the following data sets: (a) Iris [33]; (b) Wine [33]; (c) Digit1 [2]; and (d) USPS [2]

[12] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.

[13] B. Settles, M. Craven, and S. Ray, “Multiple-instance active learning,” in *In Advances in Neural Information Processing Systems (NIPS*. MIT Press, 2008, pp. 1289–1296.

[14] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, “Active learning with statistical models,” *J. Artif. Int. Res.*, vol. 4, no. 1, pp. 129–145, Mar. 1996.

[15] N. Roy and A. McCallum, “Toward optimal active learning through sampling estimation of error reduction,” in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 441–448.

[16] A. I. Schein and L. H. Ungar, “Active learning for logistic regression: an evaluation,” *Mach. Learn.*, vol. 68, no. 3, pp. 235–265, Oct. 2007.

[17] Z. Xu, R. Akella, and Y. Zhang, “Incorporating diversity and density in active learning for relevance feedback,” in *Proceedings of the 29th European conference on IR research*, ser. ECIR’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 246–257.

[18] K. Nigam, A. K. Mccallum, S. Thrun, and T. Mitchell, “Text classification from labeled and unlabeled documents using em,” in *Machine Learning*, vol. 39, 2000, pp. 103–134.

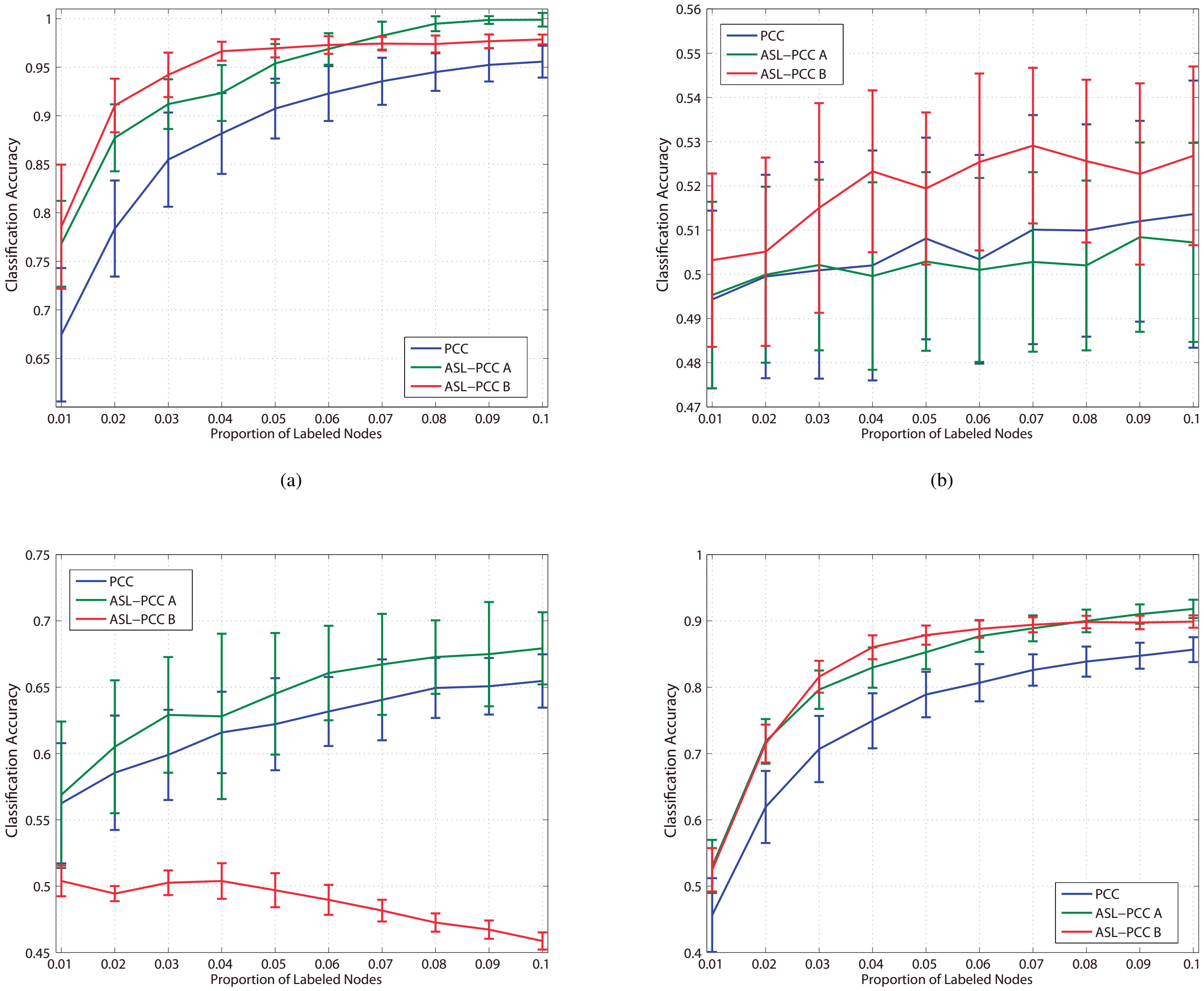
[19] A. Fujino, N. Ueda, and K. Saito, “A hybrid generative/discriminative approach to semi-supervised classifier design,” in *AAAI-05, Proceedings of the Twentieth National Conference on Artificial Intelligence*, 2005, pp. 764–769.

[20] 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.

[21] R. Dara, S. Kremer, and D. Stacey, “Clustering unlabeled data with soms improves classification of labeled real-world data,” in *Proceedings of the World Congress on Computational Intelligence (WCCI)*, 2002, pp. 2237–2242.

[22] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998, pp. 92–100.

[23] T. M. Mitchell, “The role of unlabeled data in supervised learning,” in *Proceedings of the Sixth International Colloquium on Cognitive*



(a)

(b)

(c)

(d)

Fig. 2. Correct classification rate and standard deviation comparison when the methods are applied varying the amount of labeled nodes in the following data sets: (a) COIL$_2$; (b) BCI [2]; (c) g241c [2]; and (d) COIL [2].

Science, 1999.

[24] 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.

[25] ——, "Semisupervised regression with cotraining-style algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 11, pp. 1479–1493, Nov. 2007.

[26] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley-Interscience, September 2008.

[27] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using gaussian fields and harmonic functions," in *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003, pp. 58–65.

[28] 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.

[29] 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.

[30] 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.

[31] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, and J. Liu, "Particle competition and cooperation in networks for semi-supervised learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 24, no. 9, pp. 1686 –1698, sept. 2012.

[32] F. Breve and L. Zhao, "Fuzzy community structure detection by particle competition and cooperation," *Soft Computing*, vol. 17, no. 4, pp. 659–673, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00500-012-0924-3

[33] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml