1st BRICS Countries Congress (BRICS-CCI) and 11th Brazilian Congress (CBIC) on Computational Intelligence

Semi-Supervised Learning with Concept Drift using Particle Dynamics applied to Network Intrusion Detection Data

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

Liang Zhao
Institute of Mathematics and Computer Science (ICMC)
University of São Paulo (USP)
São Carlos, Brazil
Email: zhao@icmc.usp.br

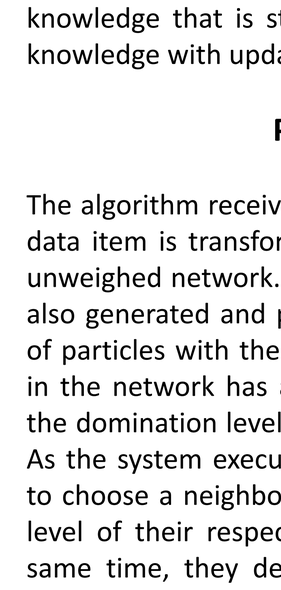
Abstract - Concept drift, which refers to non stationary learning problems over time, has increasing importance in machine learning and data mining. Many concept drift applications require fast response, which means an algorithm must always be (re)trained with the latest available data. But the process of data labeling is usually expensive and/or time consuming when compared to acquisition of unlabeled data, thus usually only a small fraction of the incoming data may be effectively labeled. Semi-supervised learning methods may help in this scenario, as they use both labeled and unlabeled data in the training process. However, most of them are based on assumptions that the data is static. Therefore, semi-supervised learning with concept drifts is still an open challenging task in machine learning. Recently, a particle competition and cooperation approach has been developed to realize graph-based semi-supervised learning from static data. We have extend that approach to handle data streams and concept drift. The result is a passive algorithm which uses a single classifier approach, naturally adapted to concept changes without any explicit drift detection mechanism. It has built-in mechanisms that provide a natural way of learning from new data, gradually "forgetting" older knowledge as older data items are no longer useful for the classification of newer data items. The proposed algorithm is applied to the KDD Cup 1999 Data of network intrusion, showing its effectiveness.

Motivation

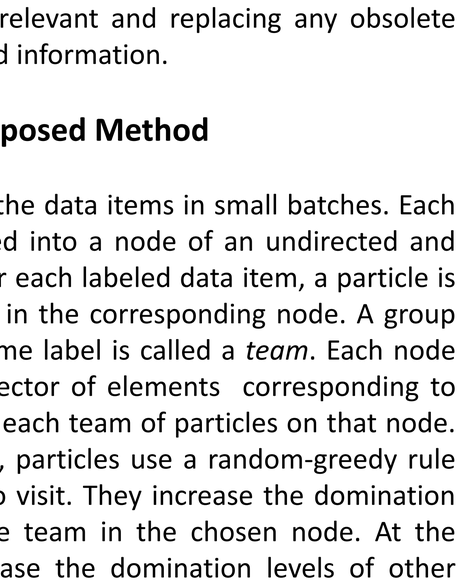
Many machine learning algorithms are designed based on the assumption that the databases are static and data samples are independent. However, in practical situations, these assumptions usually do not hold. Some examples include climate prediction, fraud detection, network intrusion, energy demand, and many other real-world applications, in which concepts and data distributions may not be stable over time. Applying traditional methods to such problems would inevitably result in low performance. In order to handle these problems, the technique has to incrementally learn from streams of data fed to it either online or in small batches. These methods have to address two conflicting objectives: retaining previously learned knowledge that is still relevant and replacing any obsolete knowledge with updated information.

Proposed Method

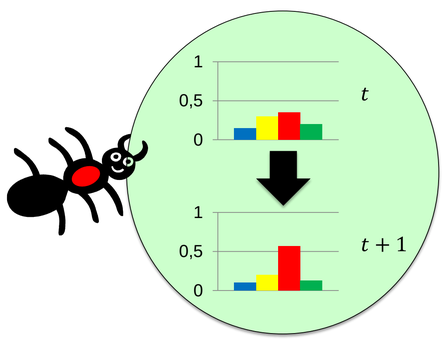
The algorithm receives the data items in small batches. Each data item is transformed into a node of an undirected and unweighed network. For each labeled data item, a particle is also generated and put in the corresponding node. A group of particles with the same label is called a *team*. Each node in the network has a vector of elements corresponding to the domination level of each team of particles on that node. As the system executes, particles use a random-greedy rule to choose a neighbor to visit. They increase the domination level of their respective team in the chosen node. At the same time, they decrease the domination levels of other teams. Each team of particles will act cooperatively trying to dominate as many nodes as possible, while preventing intrusion of other teams in their territory. Each unlabeled node will be labeled according to the team that have dominated it.



Network Formation



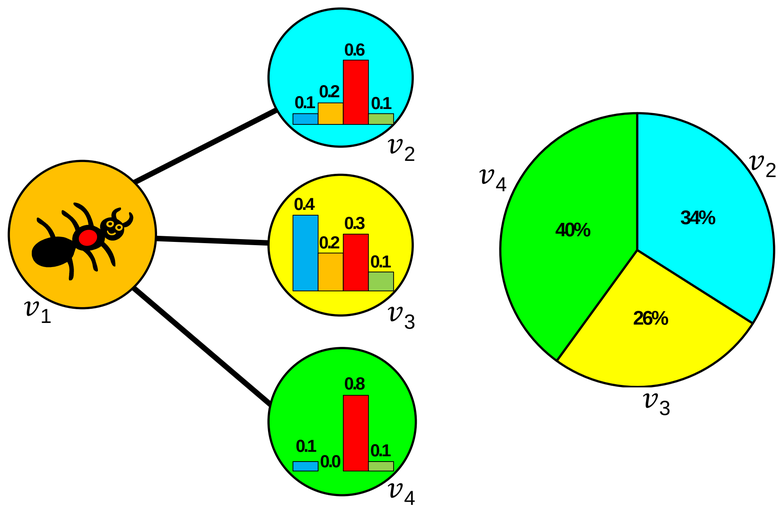
Particle Strength Changes



Particle Increasing Its Team Domination Level on the Node it is Visiting

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

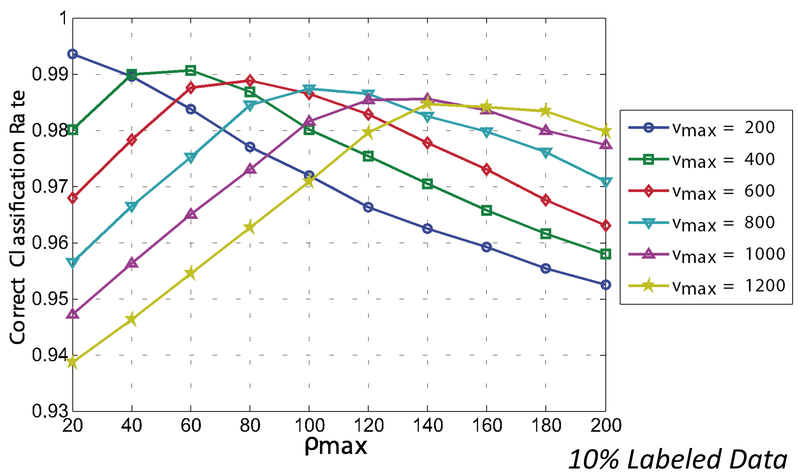
Random-Greedy Walk Probabilities Equation



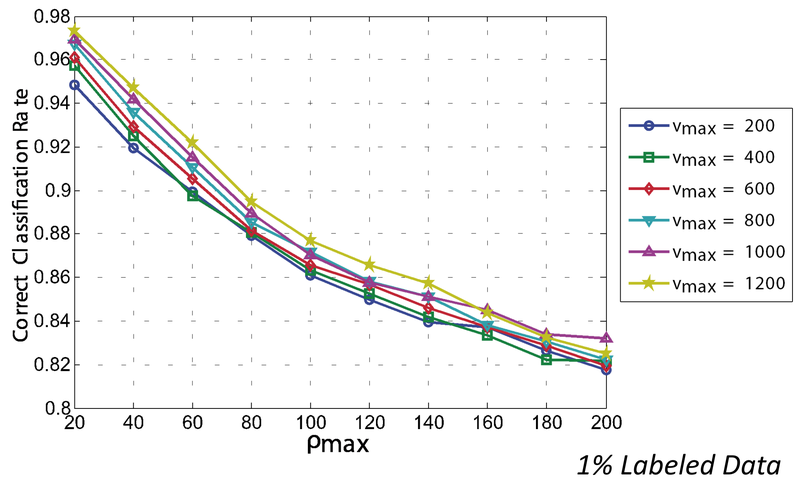
Random-Greedy Walk Moving Probabilities

When the network maximum size (v_{max}) is reached, older nodes are labeled and removed as new nodes are created. When maximum amount of particles (ρ_{max}) is reached, older particles are removed as new particles are created.

Computer Simulations



10% Labeled Data



1% Labeled Data

Conclusions

The algorithm is inspired in the competitive and cooperative behavior of some animals and the way they mark and protect their territory. It is a semi-supervised learning graph-based algorithm, which takes advantage of both labeled and unlabeled data. Each team of particles represents the labeled nodes of the same class. The particles in the same team cooperate with their teammates in order to spread their labels by walking and marking territory in the network. At the same time, each team try to expand its domain by competing against other teams. The algorithm receives small batches of data items from data streams and it is specially designed to handle gradual or incremental changes in concepts. It naturally adapts to concept changes without any explicit drift detection mechanism. Unlike other methods, it does not rely on base classifiers with explicit retraining process, it has built-in mechanisms to provide a natural way of learning from new data, gradually "forgetting" older knowledge as older labeled data items become less influent on the classification of newer data items. Different from most other passive methods, which rely on classifier ensembles, the proposed algorithm is a single classifier approach. The method can also be easily applied to online data streams. In fact, if we use batches containing only a single item, it already does that. However, the overhead of network reconfiguration would be too high. So, our next step is to generate less computational intensive ways of reconfiguring the network (recalculating the k-nearest neighbors of each node). As a future work, we also intend to build mechanisms to dynamically set the network sizes and amount of particles according to the data being fed to the algorithm. This mechanism can highly improve the performance of the algorithm in non stationary environments where the concepts may increase/decrease their evolving rhythm through time. In this sense, the algorithm would behave like an active algorithm, not by detecting individual drifts, but rather by detecting changes in drifting frequency and intensity.