An Online Algorithm for Learning a Labeling of a Graph

Kristiaan Pelckmans, Johan A.K. Suykens KRISTIAAN.PELCKMANS@ESAT.KULEUVEN.BE
SCD/sista - ESAT - KULeuven - Kasteelpark 10, 3001 Leuven, Belgium

Abstract

This short report analyses a simple and intuitive online learning algorithm - termed the *graphtron* - for learning a labeling over a fixed graph, given a sequence of labels. The contribution is twofold, (a) we give a theoretical characterization of the possible sequence of mistakes, and (b) we indicate the use for extremely large-scale problems due to sublinear space complexity and nearly linear time complexity.

This work originated from numerous discussions with John, Mark and with Johan.

1. Introduction

Many prediction problems can be reduced to the basic problem of predicting the labeling of all nodes in a given graph, after observing a few labels. We mention the application of labeling web pages as 'spam' or 'non-spam' on the www network after seeing some example pages with corresponding label, the selection of people in a social network as a potential advertisement target, or predicting disease relatedness over functional gene networks. This setting of *transductive inference* is considered as 'less complex' (in some sense) compared to the general inductive learning setting, where one not only aims for the labeling of the given nodes (datapoints), but for a generic predictive rule as well. A main advantage of studying this scheme is that one has to learn over finite domains (all possible labelings). This work explores further this learning scheme as introduced in (Vapnik, 1998) and followup work, and specifies further towards finite, weighted undirected graphs as in (Blum & Chawla, 2001; Joachims, 2003; Blum et al., 2004; Hanneke, 2006), and work done by the author (Pelckmans et al., 2006; Pelckmans et al., 2007a; Pelckmans et al., 2007b; Pelckmans et al., 2007c). A probabilistic approach was taken in the above publications, relying basically on a suitable random sampling mechanism of the labeled nodes, giving rise to firm probabilistic guarantees based on exponential concentration inequalities. Moreover, one considered here the batch learning setting, where the labels which are to be used for training are all available at the time of application of the learning algorithm.

This work takes another route, namely that of online learning, where the learning machine is presented with a sequence of labels, and has to predict those only based on the preceding labels. The classical algorithm corresponding to this scheme is the perceptron algorithm, marking the start of the movement of artificial intelligence. It is only recently that this is applied for the setting of learning over graphs, namely in (Herbster & Pontil, 2007) a modification of the perceptron is introduced, based upon the role of the (pseudo-inverse of the) graph Laplacian to represent the nodes in a genuine coordinate system, and application of the perceptron on this one follows straightforwardly. Here already, the role of the graph cut induced by the true labeling (in combination with the graph resistance diameter) was found of paramount use in the derivations.

Consider weighted undirected graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes in $\mathcal{V}$ and loopless edges $\mathcal{E}$ with positive weights $\{a_{ij} = a_{ji} \geq 0\}_{ij}$. Let the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ with $\mathbf{A}$ the positive adjancy matrix, and $\mathbf{D}$ the corresponding degree matrix. Remark that the vector 1_n belongs by construction to the null-space of $\mathbf{L}$. The *graph cut* associated to a labeling $y \in \{-1, 1\}^n$ over the nodes can then be formalized as

$$\text{cut}(y) = \sum_{y_i \neq y_j} a_{ij} = \frac{1}{4} \sum_{i,j=1}^{n} a_{ij}(y_i - y_j)^2 = \frac{1}{4} y^T \mathbf{L} y. \tag{1}$$

2. Graphtron Algorithm

Consider the *graphtron* online algorithm as described in algorithm 1, with the set M cumulating the mistakes. The proposed graphtron algorithm. Note that

Preliminary work. Under review by the International Workshop on Mining and Learning with Graphs (MLG). Do not distribute.

ties (or $\sum_{j\in M_m} a_{ij}y_j^* = 0$) are treated always as mistakes in this scheme.

Algorithm 1 Graphtron

Input: initialize $M = \{\}, m = 0$
repeat
1. An adversarial asks the label of node i.
2. We predict

$$\hat{y}_i = \text{sign}\left(\sum_{j\in M_m} a_{ij}y_j^*\right)$$

3. Nature provides the true label y_i^*

if $\hat{y}_i \neq y_i$ **then**
$M_{m+1} = M_m \cup \{i\}$ and $m = m + 1$
end if
until one is satisfied (computationally, accuracy)

This algorithm will make only a small number of mistakes, and the occurrence of mistakes can be characterized in terms of the graph topology. At first, we define the notion of the mistake subgraph $\mathcal{G}_M$ as follows:

Definition 1 (Mistake Subgraph) *Let $M = M_M$ contain the indices of nodes where the algorithm incurs a mistake. Then the mistake subgraph $\mathcal{G}_M$ is the subgraph of $\mathcal{G}$ which only contains the nodes in M, and the present edges between them. Furthermore, let d_M be the degrees of the subgraph spanned by the nodes in M, or $d_{M,i} = \sum_{j\in M} a_{ij}$.*

The analysis is much in the same style of Novikoff's mistake bound for the perceptron algorithm.

Lemma 1 (Mistake Bound) *Let y^* be the true labeling. The above algorithm will incur at most $|M|$ mistakes where*

$$\sum_{i\in M} d_{M,i} \leq 4\,\text{cut}(y^*),$$

where $\sum_{i\in M} d_{M,i}$ equals twice the weight of all edges in the mistake graph $\mathcal{G}_M$.

Proof: The proof relies on decomposing the true labeling in the mistaken labels (in the set M) and the correctly predicted ones (in the set T) such that one has

$$y^* = y_M + y_T,$$

where $y_{M,i} = y_i^*$ for $i \in M_m$ and zero otherwise, and similarly $y_{T,i} = y_i^*$ for $i \notin M_m$ and zero otherwise Let A_L denote the lower triangular part of A such that $\mathbf{A} = \mathbf{A}'_L + \mathbf{A}_L$. Now, note that the consequence of predictions made by the algorithm can be written as

$$\hat{y} = \text{sign}\left(\mathbf{A}_L y_M\right),$$

where the sign is applied elementwise. Remark that a key issue is that the diagonal of $\mathbf{A}$ are all zero, and $\mathbf{A}_L y_M$ is not dependent on the currently node. The following inequality provides the crux of the argument

$$y'_M\mathbf{L}y_M = y'_M\mathbf{D}y_M - 2y'_M\mathbf{A}_L y_M \geq y'_M\mathbf{D}y_M,$$

since $y'_M\mathbf{A}_L y_M = y'_M\mathbf{A}'_L y_M$, and $y_M(\mathbf{A}_L y_M)$ will contain only mistakes and is necessarily smaller than 0. Conversely, one has

$$4\,\text{cut}(y^*) = y'\mathbf{L}y \geq y'_M\mathbf{L}_M y_M$$

since the graph spanned by only the nodes in M is a subgraph of the total graph with Laplacian $\mathbf{L}_M$. Let $\mathbf{D}_T$ and $\mathbf{D}_M$ be the degree matrices of the after cutting the nodes in M, and the remaining ones respectively such that $\mathbf{D}_{M,ii} = \sum_{j\in M} a_{ij}$ and $\mathbf{D}_{T,ii} = \sum_{j\notin M} a_{ij}$ for all $i = 1, \ldots, n$, and $\mathbf{D} = \mathbf{D}_T + \mathbf{D}_M$. Then one has that $y'_M\mathbf{L}_M y_M = y'_M(\mathbf{D}_M - \mathbf{A})y_M$ and

$$y'_M\mathbf{L}_M y_M = y'_M(\mathbf{D}-\mathbf{D}_T-\mathbf{A})y_M = y'_M\mathbf{L}y_M - y'_M\mathbf{D}_T y_M$$

Combining the above (in)equalities yields

$$4\,\text{cut}(y^*) \geq y'_M(\mathbf{D}-\mathbf{D}_T)y_M = \sum_{i\in M}(d_i - d_{T,i})$$

since $\mathbf{D}$ and $\mathbf{D}_T$ are diagonal, and the result follows.

□

Specifically, if one has $\text{cut}(y^*) = 0$, one could not make any mistakes which are linked together, or $\sum_{i\in M} d_{M,i} = 0$ (if two nodes were connected, they could not have a different label). This inequality can now be worked out to give a specific bound for various topologies. For example, consider a binary fully connected graph (a clique). If all nodes have the same labels (or $\text{cut}(y^*) = 0$), one has $m \leq 1$ which is tight. If the clique contains two disjunct classes, one has

$$(m-1)m \leq 4\,\text{cut}(y^*)$$

since any subgraph of m nodes will have a total weight of $m(m-1)$. Similarly, if one has two disjunct cliques and labeling y^* with $\text{cut}(y^*) = 0$, one has $m \leq 2$ which again is tight. Thirdly, consider a binary weighted graph consisting of 2 cliques with a single link between those, and assume the true labeling cuts this single edge e. Then the algorithm could incur at most three mistakes (one at the interface of clique 1 with e, one inside clique 2, and the last at the intersection of clique 2 with e), and the bound would work out to be tight again.

3. Discussion

We enumerate some strengths of the algorithm.

1. The time complexity is at most $O(nm)$ if one could identify the links from any point to the m points in M in $O(m)$. In case the number of mistakes m is $O(1)$, the time complexity requirement is linear. This is a considerable improvement over the approach proposed in (Herbster & Pontil, 2007) requiring the computation of the pseudo-inverse of the graph Laplacian.

2. The space requirement is only $O(m)$. and there is no need whatsoever to store the full graph at a single instance in memory. This makes this learning algorithm especially useful for learning over growing graphs.

3. We do not rely on any instance on an appropriate, random sampling scheme. From the analysis it even follows that one would benefit largely by scheduling the nodes incurring a mistake as soon as possible. This makes this approach especially appropriate for experimental design and explorative settings.

4. The algorithm appeals to intuition in that one only learns from (and memorizes) nodes whose labels do not match ones expectation from looking at previous experience. This arguably matches the dynamics of education fairly well - as it is the task of the teacher to show how knowledge can be improved (or a question/node will be mispredicted by the student).

A practical validation of the scheme will be presented in the full publication, as well as various nontrivial bounds of the term $\sum_{i \in M} d_{M,i}$ for various topologies.

References

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the eighteenth international conference on machine learning (icml)*, 19–26. Morgan Kaufmann Publishers.

Blum, A., Lafferty, J., Rwebangaria, M., & Reddy, R. (2004). Semi-supervised learning using randomized mincuts. In *Proceedings of the eighteenth international conference on machine learning (icml)*. Morgan Kaufmann Publishers.

Hanneke, S. (2006). An analysis of graph cut size for transductive learning. *In proceedings of the 23rd International Conference on Machine Learning (ICML)*..

Herbster, M., & Pontil, M. (2007). Prediction on a graph with a perceptron. In B. Schölkopf, J. Platt and T. Hoffman (Eds.), *Advances in neural information processing systems 19*, 577–584. Cambridge, MA: MIT Press.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *International Conference on Machine Learning (ICML)* (pp. 290–297).

Pelckmans, K., Shawe-Taylor, J., Suykens, J., & De Moor, B. (2007a). Margin based transductive graph cuts using linear programming. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, (AISTATS 2007), pp. 360-367*. San Juan, Puerto Rico.

Pelckmans, K., Suykens, J., & De Moor, B. (2007b). Transductive learning over graphs: Incremental assessment. *International The Learning Workshop (SNOWBIRD), Technical Report ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2007-06*. San Juan, Puerto Rico.

Pelckmans, K., Suykens, J., & Moor, B. D. (2006). The kingdom-capacity of a graph: On the difficulty of learning a graph labelling. In *in proc. of the workshop on machine learning on graphs*, 1–8. Berlin, Germany: TBA.

Pelckmans, K., Suykens, J., & Moor, B. D. (2007c). Transductive rademacher complexities for learning over a graph. In *The 5th international workshop on mining and learning with graphs*, 1–8. Firenze, Italy.

Vapnik, V. (1998). *Statistical learning theory*. Wiley and Sons.