Transductively Learning from Positive Examples Only

Kristiaan Pelckmans and Johan A.K. Suykens [*]

K.U.Leuven - ESAT - SCD/SISTA, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium

Abstract. This paper considers the task of learning a binary labeling of the vertices of a graph, given only a small set of positive examples and knowledge of the desired amount of positives. A learning machine is described maximizing the precision of the prediction, a combinatorial optimization problem which can be rephrased as a S-T mincut problem. For validation, we consider the movie recommendation dataset of MOVIELENS [1]. For each user we have given a collection of (ratings of) movies which are liked well, and the task is to recommend a disjoint set of movies which are most probably of interest to the user.

1 Introduction

Machine learning provides a rich framework to study prediction algorithms tailored to a task at hand. This short paper discusses some aspects of the learning task where only a set of positive labels are to be used to make predictions. In particular, the setting is adopted where we have to qualify rules which can be used for predicting whether an instance is relevant to a specific situation, or not. We consider the example of designing a movie recommender system. Here we have given a finite collection of movies. For a given user we have to predict which movies can be expected to please the customer, given a set of movies he has seen already. Now, an important realization [2] in this context is that the customer has selected his/her previous seen movies based on an expectation that he/she will have liked them. A customer wouldn't have bothered to endure (or rate) a movie which he expected (at the time) to be annoyed by. This mechanism of selecting labels makes the nature of the task quite different from classical statistical learning settings, where the sampling may be assumed to follow a random scheme (e.g. i.i.d.). Secondly, a recommender system is not really interested in the number of mistaken predictions it will make on the full collection of movies. At the end of the day, one is only interested how well the recommendations worked out. In particular, ti would be worse to recommend positively a 'non-interesting' movie, rather than to rate a potentially interesting movie as negative.

[*] KP is a post doctoral researcher with the FWO Flanders, Belgium, and JS is a professor at the Katholieke Universiteit Leuven, Belgium. Research supported by (Research Council KUL) GOA AMBioRICS, CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants; (Flemish Government FWO) PhD/postdoc grants, projects G.0452.04 (new quantum algorithms), G.0499.04 (Statistics), G.0211.05 (Nonlinear), G.0226.06 (cooperative systems and optimization), G.0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, McKnow-E, Eureka-Flite+ (Belgian Federal Science Policy Office) IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011) ; (EU) ERNSI; FP7-HD-MPC (Collaborative Project STREP-grantnr. 223854) Contract Research: AMINAL; Other: Helmholtz: viCERP

[1] This dataset can be found at http://www.grouplens.org/

[2] 'Three levels of addressing the Netflix Prize', see http://hunch.net/?p=331

A challenging and inspiring conjecture issued in the context of machine learning is that unlabeled data can help solving supervised learning problems. Albeit this question is still open theoretically (in general), this working hypothesis led already to the development of successful practical algorithms, an exciting subfield which is surveyed in [1, 2]. Transductive inference concerns a related problem where we restrict attention to predicting the labels of the given unlabeled examples. This approach provides an appropriate context for our learning task since we will be concerned with selecting a good candidate from a finite collection: there is no need for constructing a preference function which can be evaluated on infinitely many subjects. We refer to [3, 4, 5] and citations for details. A little thought reveals that our learning setting can be phrased as a problem of selective inference [3]: ”given a collection of objects and a finite collection of corresponding labels, find unlabeled objects which can be predicted most accurately”. This learning setting is conjectured to be even simpler than the transductive case, but is not studied in detail due to (amongst others) lack of an efficient algorithm. This paper analyses a technique which will implement a specific form of selective inference which can be used for the recommendation setting. Our precise setting however is still different in that the set of observed labels is not drawn randomly (or i.i.d.). It is exactly this question we will try to shed some insight in in the following.

Some notation is introduced. Let a weighted undirected graph $\mathcal{G}_n = (\mathcal{V}, \mathcal{E})$ consist of $1 < n < \infty$ nodes $\mathcal{V} = \{v_i\}_{i=1}^n$ with edges $E = \{e_{ij}\}_{i \neq j}$ having weights $w(e_{ij}) = a_{ij} \geq 0$ for any $i \neq j = 1, \ldots, n$. Assume that no loops occur in the graph, i.e. $a_{ii} = 0$ for all $i = 1, \ldots, n$. Let $A \in \mathbb{R}^{n \times n}$ denote the positive symmetric matrix defined as $A_{ij} = A_{ji} = a_{ij}$ for all $i, j = 1, \ldots, n$. The Laplacian of $\mathcal{G}$ is then defined as $\mathbf{L} = \operatorname{diag}(A1_n) - A \in \mathbb{R}^{n \times n}$. This paper considers problems where each node v_i has a fixed corresponding label $y(v_i) \in \{-1, 1\}$ but only a subset $\mathcal{S}_m \subset \{1, \ldots, n\}$ with $|\mathcal{S}_m| = m$ of the labels is observed. The task in transductive inference is to predict the labels of the unlabeled nodes $\mathcal{S}_m^- = \{1, \ldots, n\} \backslash \mathcal{S}_m$. Let a function $q : \mathcal{V} \to \{-1, 1\}$ denote a hypothesis. We will alternatively and interchangeably use the vector notation $q_n \in \{-1, 1\}^n$ where $q_{n,i} = q(v_i)$.

This paper assumes that an underlying *true* labeling y exists (but is unknown). This simplification will simplify the exposition considerably, and extension to the case where the observed output is a random variable itself can be obtained using standard results (at least computationally). A second important assumption is that the *complexity* of q (i.e. an intuitive measure of how plausible q is), is measured by how many edges connect the subgraphs corresponding to $+1$ and -1 labeled vertices. From the above definitions, it follows that the graphcut associate to a graph $\mathcal{G}$ and a labeling q of the vertices can be written as

$$\operatorname{cut}(q) = \sum_{q(v_i) \neq q(v_j)} a_{ij} = \frac{1}{4} q_n^T \mathbf{L} q_n. \tag{1}$$

This short paper is organized as follows. Section 2 describes the formal learning setting and illustrates how one can implement risk minimization by a min cut- max flow algorithm. Section 3 gives some insight in the practical use of this technique applied on the movielens recommendation task.

2 Learning from Positive Labels

2.1 Learning Setting

Next, we spend some time on formalizing the precise objective of our learning task. The transductive risk term is given as

$$\mathcal{R}(q) = P\left(y(V) \neq q(V)\right), \tag{2}$$

where V denote a randomly selected node $V \in \mathcal{V}$ and its corresponding label $y(V) \in \{-1, 1\}$. Given the labels of a random subset of $\mathcal{S}_m \subset \mathcal{V}$, its empirical counterpart becomes $\mathcal{R}_m(q, \mathcal{S}_m) = \frac{1}{m}\sum_{i\in\mathcal{S}_m} I(q(v_i) \neq y(v_i))$, and the test error is $\mathcal{R}_m^-(q, \mathcal{S}_m^-) = \frac{1}{n-m}\sum_{i\notin\mathcal{S}_m} I(q(v_i) \neq y(v_i))$. Here the empirical risk term can serve as a proxy to the (unknown) $\mathcal{R}(q)$ whenever the sampling follows a random sampling scheme. This is the case when the different vertices which are labeled are independently sampled from an underlying, fixed distribution (as in [3]), or when the samples are uniform without replacement (as in transductive inference, see e.g. [4] and citations).

In our learning setting, we argue that a more natural objective in recommender systems would be the rate of false positives. The *precision* amounts to the number of actual positives amongst the positive predictions. This measure is equal to one minus the False Discovery Rate (FDR). In the movie recommender context, this amounts to the number of recommendations actually enjoyed by a customer who followed the systems recommendation. Let $\mathcal{S}_q^+ = \{i : q(v_i) = 1\}$. Formally the precision $Pre(q) \in [0, 1]$ of a result q is defined as

$$Pre(q) = P(q(V) = y(V) \mid q(V) = 1) = \frac{1}{|\mathcal{S}_q^+|}\sum_{i\in\mathcal{S}_q^+} I(y(v_i) = 1), \tag{3}$$

where the probability concerns a uniformly randomly selected node $V \in \mathcal{V}$ given $q(V) = 1$. We adopt the convention that $Pre(q) = 0$ if $\mathcal{S}_q^+ = \{\}$. Since we cannot evaluate all values of $y(v_i)$ with $v_i \in \mathcal{S}_q^+$, it appears hard to evaluate this quantity or to estimate it from a final sample set. We do however want to penalize the number of nodes with positive label predicted as being negative, and the size $|\mathcal{S}_q^+|$. On the other hand, we have the *recall* qualifies how many actual positives are recover by the prediction rule q. In the setting of our recommender system that would be the probability of the system recommending a movie which should really be advertised.

$$Rec(q) = P\left(q(V) = y(V) \mid y(V) = 1\right). \tag{4}$$

Given a set of positively labeled vertices $\mathcal{S}_m^+ \subset \{v_i : y(v_i) = 1\}$, its empirical counterpart becomes $Rec_m(q) = \frac{1}{m}\sum_{i\in\mathcal{S}_m^+} I(q(v_i) = 1)$. Now one sees that $Rec_m(q)$ approximates $Rec(q)$ when the observed labels S_m is a random subset from $\mathcal{S}_y^+ = \{v \in \mathcal{V} : y(v) = 1\}$. It should be remarked that this set $\mathcal{S}_y^+$ is deterministic only if y is fixed, and relaxing the labeling to be random as well will influence a theoretical analysis considerably.

2.2 Reformulation as a MINCUT problem

When designing a learning machine, it is paramount to think on an appropriate collection of possible solutions - or a *hypothesis space*. When the data is organized in a directed (weighted) graph, a natural choice is to consider labelings which separates parts of the class which are not too strongly connected as characterized by the graphcut of a labeling (see equation 1).

$$\mathcal{H}_\rho = \Big\{ q_n \in \{-1, 1\}^n : q_n^T \mathbf{L} q_n \leq \rho \Big\}. \tag{5}$$

In [5], it was shown how the cardinality of this class can be bounded in terms of the eigenvalue spectrum of **L**. Now, the one minimizing the rate of false negatives is given by the following optimization problem.

$$\hat{q} = \arg\max_{q_n^T \mathbf{L} q_n \leq B} Pre(q). \tag{6}$$

There are however a number of problems with this formulation. (A) there is a trivial solution where $q_n = 1_n$, moreover there is no direct way to manipulate how many negative predictions one should make. (B) the precision $Pre(q)$ cannot be estimated directly, and (C) for many values of B the solution is not unique, and values for which the optimum is unique are entirely data-dependent. Therefor, we consider a slightly different formulation

$$\hat{q} = \arg\min_q \frac{1}{4} q_n^T \mathbf{L} q_n \quad \text{s.t.} \quad \begin{cases} \sum_{i=1}^n I(q(v_i) = 1) \leq \rho & \\ q(v_i) = 1 & \forall v_i \in \mathcal{S}_m^+ \end{cases} \tag{7}$$

where precision $Pre(q)$ improves when ρ is taken smaller. This problem formulation has the problem that (A) the parameter ρ is not often known in advance; (B) observed positive labels can be mistaken or are not typical (disconnected to the set of nodes of interest). For trade-off parameters $\gamma \geq 0$ and $\lambda \geq 0$, the algorithm under consideration implements the following optimization problem.

$$\hat{q} = \arg\min_q \frac{1}{4} q_n^T \mathbf{L} q_n + \gamma \sum_{i \in \mathcal{S}_m^+} I(q(v_i) = -1) + \lambda \sum_{i=1}^n I(q(v_i) = 1) \tag{8}$$

One can consider the terms γ and λ as the Lagrange multipliers corresponding to the constaints in (7) of the hard constraint ρ in $\mathcal{H}_\rho$. As in [6], this combinatorial optimization problem can be implemented efficiently as follows.

1. Extend $\mathcal{G}$ with two vertices v^- and v^+.
2. Connect the vertices with given positive labels with v^+ with weight γ.
3. Connect all n nodes with v^-, with weight λ.
4. Find a min S-T cut between source v^- and sink v^+.

5. Assign to the nodes $v \in \mathcal{V}$ still connected to v^+ a positive label, and to the remaining ones a label -1.

This can be seen by simply counting the total weight of the edges in the extended graph which are cut by an optimal solution, a number which will correspond with the value of the objective eq. (8). Observe that we will obtain a trivial solution $q_n = -1_n$ if $\lambda \geq \gamma$, and useful solutions are typically obtained by choosing $\lambda \ll \gamma$. The algorithm for calculating the minimal cut between a Source and Sink node lies at the core of combinatorial optimization, see e.g. [7, 8]. the push-relabel maximum flow algorithm has a time complexity of $O(n^2|E|)$ where $|E|$ denotes the cardinality of the set of nonzero edges. It is also classical that the above problem can be solved by a convex linear program, a technique which was employed in [5].

3 Experiments

In order to illustrate the technique, we employ the implementation of the push-relabel algorithm for computing the min S-T cut as given by MatlabBGL [3], in turn using the Boost Graph Library [4]. At first, we conduct an artificial example, described in Figure 1. Secondly, we consider a simple experiment on the MOVIELENS recommendation task. The data consist on the preferences of 943 different users, each giving ratings on some of the IDs of 1682 movies. For each user, we consider the task of recommending 4+ rated movies, given a subset of movies rated by this use previously as 4+. This task is repeated for all users, and the precision and recall of each user-specific recommendation is computed. The design of the graph in which the movies are organized is paramount. We found empirically that computing the weight of an edge $a_{ij} = \lfloor \ln(\#(M_i, M_j)) \rfloor$ for all movies M_i and M_j works well. Here $\#(M_i, M_j)$ equals the number of users in the dataset both rating the 2 movies M_i and M_j as 4+, organizing almost 75% of the movies in a connected graph. We found precisions until almost 20%. The average precision precision is 8.45%, and the average recall is 16.98%, averaged over the 943 users. A naive algorithm recommending the 10 movies closest connected in the graph to the given positively labeled vertices, yields a precision of 6.12% and an average recall of 10.01%.

4 Conclusion

This short paper discussed the learning task of recommending objects which are also positive, given a collection of purely positively labeled objects. We gave some insight into the nature of the learning problem, and its comparison to transductive and selective inference. The resulting combinatorial optimization problem was found to be solvable by using a min cut - max flow algorithm. Results on the MOVIELENS recommendation dataset are given. A most interesting question still is how one can validate (model selection) a recommender system. This issue is approached here using observational data, but the usefulness of a recommender system should be measured really online (and actively).

[3] http://www.stanford.edu/d̃gleich/programs/matlab_bgl/

[4] http://www.boost.org/

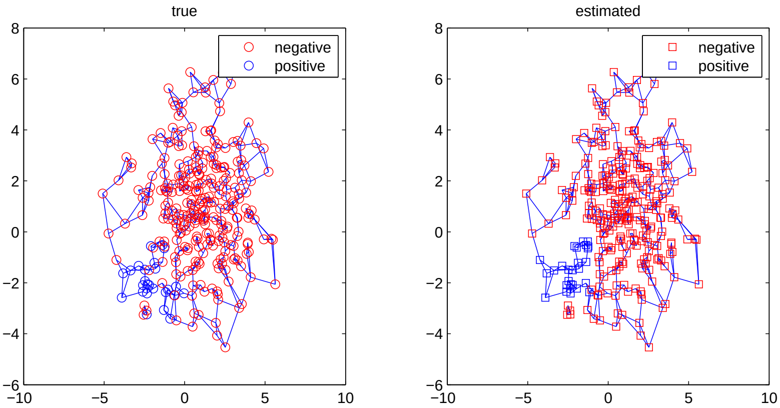

Fig. 1: *Artificial example with 280 negative samples (red) and 20 positives (blue). Only 3 positively labeled vertices are given to the learning algorithm. The graph was obtained by a 2-nearest neighbor rule based on the position of the 300 nodes sampled from* $\mathcal{N}((-2,-2),I_2)$ *and* $\mathcal{N}((2,2),I_2)$ *corresponding with the two classes. (a) the true labels, and (b) the estimated labels. This precise example achieves a precision of* 69% *and a recall of* 72%.

References

[1] Xiaojin Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Melon University, May 2005. CMU-LTI-05-192.

[2] O. Chapelle, B. Schölkopf, and A. Zien(Eds.), editors. *Semi-supervised Learning*. MIT Press, Cambridge, MA, 2006.

[3] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, 2006.

[4] R. El-Yaniv P. Derbeko and R. Meir. Explicit learning curves for transduction and application to clustering and compression algorithms. *Journal of Artificial Intelligence Research*, 22:117–142, 2004.

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

[6] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 19–26. Morgan Kaufmann Publishers, 2001.

[7] J. Kleinberg and E. Tardos. *Algorithmical Design*. Addison-Wesley, 2005.

[8] Chekuri S., Goldberg A., Karger D., Levine M., and Stein C. Experimental study of minimal cut algorithms. In *Proceedings of the 8th ACM SIAM Symp. on Discr. Algorithms (SODA97)*, 1997.