Possible and necessary labels in K-nn procedures to query partially labelled data

Vu-Linh Nguyen, Sébastien Destercke, Marie-Hélène Masson

Possible and necessary labels in K-nn procedures to query partially labelled data

Vu-Linh Nguyen and **Sébastien Destercke** and **Marie-Hélène Masson**[1]

Abstract. When learning from partially labelled data (i.e., given as a subset of possible labels containing the true one), an issue that naturally arise is to determine which data should be queried to improve the accuracy of predictions, or in other word to determine an order between the partial labels to be queried. An answer to that question is to query the data that would induce the highest number of ambiguous predictions. In the K-nn case, studied in this paper, this question is similar to determining possible and necessary winners in plurality voting. In this paper, we discuss this connection as well as its use in partial label querying.

1 Introduction

The problem of learning from partial labels is known under various names: "learning with partial labels" [2], "learning with ambiguous labels" [3] or "superset label learning" [5]. In these works, authors have either proposed general schemes to learn from partial labels, for instance by adapting the loss function to partial labels [2], or to adapt specific algorithms (e.g. K-nn or decision trees) to the case of partial labels [3].

In general, the less partial are the labels, the better these techniques will perform. In the spirit of active learning techniques, this work addresses the problem of finding which partial labels should be disambiguated by an oracle (expert) in order to achieve better performances. In this work, we adopt a robust view consisting in considering all possible replacements of the partial labels instead of making any posterior probability of true labels.

In order to find which instance to query, we propose in Section 2 a general scheme based on measuring the potential impact of knowing the true label of a given instance. We then propose a specific measure to assess this impact which considers whether a partial label introduces some ambiguity in the decision, using some notions issued from social choice theory [6] in Section 3.

2 General scheme

In our setting, we assume that we have one training set $\mathbf{D} = \{(\mathbf{x}_n, \mathbf{y}_n)|n = 1, \ldots, N\}$ used to make predictions, with $\mathbf{x}_n \in \mathbb{R}^P$ the features and $\mathbf{y}_n \subseteq \Omega = \{\lambda_1, \ldots, \lambda_M\}$ potentially imprecise labels. As usual when working with partial labels [2], we assume that $\mathbf{y}_n$ contains the true label. We also assume that we have one unlabelled target set $\mathbf{T} = \{\mathbf{t}_j|j = 1, \ldots, T\}$ that will be used to determine the partial labels to query and can be defined differently based on the usage purposes as point out latter in Section 4.

[1] UMR CNRS 7253 Heudiasyc, University of Technology of Compiegne, email:{linh.nguyen, sebastien.destercke, mylene.masson}@hds.utc.fr

When data $\mathbf{y}_n$ are precise, the decision $h(\mathbf{t})$ taken by the K-nn procedure is given by

$$h(\mathbf{t}) = \arg\max_{\lambda \in \Omega} \sum_{\mathbf{x}_k^t \in \mathbf{N_t}} w_k^t \mathbb{1}_{\lambda = \mathbf{y}_k^t} \tag{1}$$

where $\mathbf{N_t} = \{\mathbf{x}_1^t, \ldots, \mathbf{x}_K^t\}$ and $\mathbf{w_t} = \{w_1^t, \ldots, w_K^t\}$ are the K nearest neighbours of $\mathbf{t}$ and their associated weights, respectively. Equation (1) directly extends to the partial label case [3] by replacing $\lambda = \mathbf{y}_k^t$ by $\lambda \in \mathbf{y}_k^t$, however this comes down to consider a particular replacement of partial labels.

This is useful when the goal is to determine a single prediction from partial labels, however if the aim is to identify those partial labels that would be the most useful to query, it seems preferable to identify which data in $\mathbf{D}$ makes predictions in $\mathbf{T}$ ambiguous, with the idea that the more ambiguity they induce, the more interesting it is to know their label.

In the next section, we discuss the problem of determining whether $\mathbf{D}$ induces some ambiguity on a particular instance $\mathbf{t}$, and also a way to quantify whether querying a particular instance $\mathbf{x}_n$ would have an effect on this ambiguity. We will see that this is strongly connected to issues from social choice theory [6].

3 Indecision-based querying criteria

In this section, we present an effect score that is based on whether a partial labelled instance $\mathbf{x}_n$ introduces some ambiguity in the decision about an instance $\mathbf{t}$. We will first define what we mean by ambiguity.

3.1 Ambiguous instance: definition

In the K-nn algorithm, each neighbour can be seen as a (weighted) voter in favor of his preferred class. Partial labels can then be assimilated to voters providing incomplete preferences. For this reason, we will define ambiguity by using ideas issued from majority voting procedure with incomplete preferences [4]. More precisely, we will use the notions of necessary and possible winners of such a voting procedure to determine when a decision is ambiguous.

For an instance $\mathbf{t}$ with $\mathbf{N_t} = \{\mathbf{x}_1^t, \ldots, \mathbf{x}_K^t\}$, we will denote by $\mathbf{L_t} = \{(l_1^t, \ldots, l_K^t)|l_k^t \in \mathbf{y}_k^t\}$ the set of possible selections of $\mathbf{N_t}$ with cardinality $|\mathbf{L_t}| = \prod_{k=1}^K |\mathbf{y}_k^t|$. For a given selection $\mathbf{l}^t \in \mathbf{L_t}$, the corresponding winner(s) of the voting procedure is (are)

$$\widehat{\lambda}_{\mathbf{l}} = \arg\max_{\lambda \in \Omega} \sum_{k=1}^{K} w_k^t \mathbb{1}_{l_k^t = \lambda}$$

with w_k^t the weight corresponding to the kth neighbor. Let us note that the $\arg\max$ can return multiple labels (we do not break ties).

We can now define the possible ($\mathbf{PL_t}$) and necessary label sets ($\mathbf{NL_t}$) of $\mathbf{t}$ as follows:

$$\mathbf{PL_t} = \{\lambda \in \Omega | \exists \mathbf{l}^t \in \mathbf{L_t} \text{ s.t } \lambda \in \widehat{\lambda}_{\mathbf{l}^t}\} \quad (2)$$

and

$$\mathbf{NL_t} = \{\lambda \in \Omega | \forall \mathbf{l}^t \in \mathbf{L_t}, \lambda \in \widehat{\lambda}_{\mathbf{l}^t}\}, \quad (3)$$

which are nothing else but the set of possible and necessary winners in social choice theory. By definition, we have $\mathbf{NL_t} \subseteq \mathbf{PL_t}$. Given a target instance, we adopt the following definition of ambiguity.

Definition 1. *A target instance* $\mathbf{t}$ *is called ambiguous if* $\mathbf{NL_t} \neq \mathbf{PL_t}$.

The ideal situation is to have $\mathbf{PL_t} = \mathbf{NL_t}$ and $|\mathbf{PL_t}| = 1$, since in this case the decision is uniquely defined.

3.2 Ambiguous instance: computational issues

Let us first provide some definitions. For each $\lambda \in \Omega$, we define the minimum and maximum scores as

$$S^{\min}(\lambda) = \sum_{k=1}^{K} w_k^t \mathbb{1}_{\lambda = \mathbf{y}_k^t} \text{ and } S^{\max}(\lambda) = \sum_{k=1}^{K} w_k^t \mathbb{1}_{\lambda \in \mathbf{y}_k^t},$$

whose computation can be done in linear time.

Computing $\mathbf{NL_t}$ The problem of determining $\mathbf{NL_t}$ is actually very easy, as it is known [4] that

$$\mathbf{NL_t} = \left\{\lambda |\ S^{min}(\lambda) \geq S^{max}(\lambda^{'}), \forall \lambda^{'} \neq \lambda, \lambda^{'} \in \Omega\right\}$$

Computing $\mathbf{PL_t}$ Determining $\mathbf{PL_t}$ is in practice much more difficult. In the unweighted case (all weights in $\mathbf{w_t}$ equals), known results indicate [1, 7] that $\mathbf{PL_t}$ can be determined in cubic (hence polynomial) time, solving a maximum flow problem and using the fact that when votes are (made) unitary, the solution of this flow problem is integer-valued (due to the submodularity of the constraint matrix).

However, when votes are non-unitary, or when weights are different, this result does not hold anymore, and the problem appears to be NP-hard. In addition to that, in our setting we can have to evaluate $\mathbf{PL_t}$ a high number of times (in contrast with what happens in social choice, where $\mathbf{PL_t}$ and $\mathbf{NL_t}$ have to be evaluated at most a few times), hence even a cubic algorithm may have a prohibitive computational time. A computationally cheap approximation is then to consider the set

$$\mathbf{APL_t} = \left\{\lambda | S^{max}(\lambda) \geq \max_{\lambda^{'} \in \Omega_{\mathbf{t}}} S^{min}(\lambda^{'}), \forall \lambda^{'} \neq \lambda, \lambda^{'} \in \Omega\right\}$$

3.3 Effect of a query on ambiguous instances

Once we know which predictions are ambiguous, it remains to determine which instances in $\mathbf{D}$ we should query in order to reduce the most this ambiguity. We adopt a simple scheme to do that: for an instance $\mathbf{x}_n$, we determine a local score $f_{\mathbf{x}_n}(\mathbf{t})$ determining whether querying $\mathbf{x}_n$ can reduce our ambiguity on $\mathbf{t}$, and then aggregate this local score into a global score

$$f_{\mathbf{x}_n}(\mathbf{T}) = \sum_{\mathbf{t} \in \mathbf{T}} f_{\mathbf{x}_n}(\mathbf{t}), \quad (4)$$

over the whole set $\mathbf{T}$ which is simply the sum of local scores over each instance t. Let us denote by $\mathbf{NL}_{\mathbf{t}}^{q_n=\lambda}$, $\mathbf{APL}_{\mathbf{t}}^{q_n=\lambda}$ and $\mathbf{PL}_{\mathbf{t}}^{q_n=\lambda}$ the sets obtained if we learn $\mathbf{y}_n = \lambda$. Then we can define the local scores

$$f_{\mathbf{x}_n}^{APL}(\mathbf{t}) = \begin{cases} 1 \text{ if } \exists \lambda \text{ s.t. } \mathbf{NL}_{\mathbf{t}}^{q_n=\lambda} \neq \mathbf{NL_t} \text{ or } \mathbf{APL}_{\mathbf{t}}^{q_n=\lambda} \neq \mathbf{APL_t}, \\ 0 \text{ else.} \end{cases}$$

and

$$f_{\mathbf{x}_n}^{PL}(\mathbf{t}) = \begin{cases} 1 \text{ if } \exists \lambda \text{ s.t. } \mathbf{NL}_{\mathbf{t}}^{q_n=\lambda} \neq \mathbf{NL_t} \text{ or } \mathbf{PL}_{\mathbf{t}}^{q_n=\lambda} \neq \mathbf{PL_t}, \\ 0 \text{ else.} \end{cases}$$

with $f_{\mathbf{x}_n}^{PL}(\mathbf{t})$ being more complex to estimate than $f_{\mathbf{x}_n}^{APL}(\mathbf{t})$, for similar reasons as the one mentioned in Section 3.2. In particular, we can show that there are easy ways to estimate $f_{\mathbf{x}_n}^{APL}(\mathbf{t})$.

We can then use any of these functions to determine the global score $f_{\mathbf{x}_n}(\mathbf{T})$, and which instance $\mathbf{x}_n$ to query.

4 Ongoing works

Our current work concerns the investigation of computational issues as well as experimental comparisons of different approaches:

- we are currently comparing different querying schemes to the use of $f_{\mathbf{x}_n}^{PL}(\mathbf{t})$ and $f_{\mathbf{x}_n}^{APL}(\mathbf{t})$, such as classical active learning, random querying, querying the most partial instances first, ... in order to know whether identifying ambiguous situations, which are computationally more difficult to identify, is beneficial to the learning procedure. Current results show that, indeed, there is an advantage in identifying those instances that induce a lot of ambiguity;
- we are also currently investigating the computational problem of determining $\mathbf{PL_t}$ in the weighted case. First results indicate that the problem is NP-hard (it seems to be reducible to a 3-dimensional matching problem), yet a refined complexity analysis is necessary to identify whether it is an important issue for our case.

REFERENCES

[1] Nadja Betzler and Britta Dorn, 'Towards a dichotomy for the possible winner problem in elections based on scoring rules', *Journal of Computer and System Sciences*, **76**(8), 812–836, (2010).

[2] Timothee Cour, Ben Sapp, and Ben Taskar, 'Learning from partial labels', *The Journal of Machine Learning Research*, **12**, 1501–1536, (2011).

[3] Eyke Hüllermeier and Jürgen Beringer, 'Learning from ambiguously labeled examples', *Intelligent Data Analysis*, **10**(5), 419–439, (2006).

[4] Kathrin Konczak and Jérôme Lang, 'Voting procedures with incomplete preferences', in *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20. Citeseer, (2005).

[5] Liping Liu and Thomas Dietterich, 'Learnability of the superset label learning problem', in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1629–1637, (2014).

[6] Hervé Moulin, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia, *Handbook of Computational Social Choice*, Cambridge University Press, 2016.

[7] Lirong Xia and Vincent Conitzer, 'Determining possible and necessary winners given partial orders', *Journal of Artificial Intelligence Research*, 25–67, (2011).