Efficient Human Computation: the Distributed Labeling Problem

Keywords: supervised learning, theory

Ran Gilad-Bachrach RAN.GILAD-BACHRACH@INTEL.COM
Intel Research Israel Lab

Aharon Bar-Hillel AHARON.BAR-HILLEL@INTEL.COM
Intel Research Israel Lab

Liat Ein-Dor LIAT.EIN-DOR@INTEL.COM
Intel Research Israel Lab

Abstract

Collecting large labeled data sets is a laborious and expensive task, whose scaling up requires division of the labeling workload between many teachers. When the number of classes is large, miscorrespondences between the labels given by the different teachers are likely to occur, which, in the extreme case, may reach total inconsistency. In this study we describe how globally consistent labels can be obtained, despite the absence of teacher coordination, and discuss the possible efficiency of this process in terms of human labor. We define a notion of label efficiency, measuring the ratio between the number of globally consistent labels obtained and the number of labels provided by distributed teachers. We show that the efficiency depends critically on the ratio α between the number of data instances seen by a single teacher, and the number of classes. We suggest several algorithms for the distributed labeling problem, and analyze their efficiency as a function of α. In addition, we provide an upper bound on label efficiency for the case of completely uncoordinated teachers, and show that efficiency approaches 0 as the ratio between the number of labels each teacher provides and the number of classes drops (i.e. $\alpha \to 0$).

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

As applications of machine learning mature, larger training sets are required both in terms of the number of training instances and the number of classes considered. In recent years we have witnessed this trend for example in vision related tasks such as object class recognition or detection (Griffin et al., 2007; Everingham et al., 2007; Russell et al., 2005). Specifically for object class recognition, current data sets such as the Caltech-256 (Griffin et al., 2007) include tens of thousands of images from hundreds of classes. Collecting consistent data sets of this size is an intensive and expensive task. Scaling up naturally leads to a distributed labeling scenario, in which labels are provided by a large number of weakly coordinated teachers. For example, in the Label-me system (Russell et al., 2005) the labels are contributed by dozens of researchers, while in the ESP game (von Ahn, 2006) labels are supplied by thousands of uncoordinated players.

As we turn toward distributed labeling, several practical considerations emerge which may disrupt the data integrity. In general, while it is reasonable to believe that a single teacher is relatively self-consistent (though not completely error-free), this is not the case with multiple uncoordinated teachers. Different teachers may have differences in their labeling systems due to several causes. First, different teachers may use different words to describe the same item class. For example, one teacher may use the word "truck" while the other uses "lorry" to describe the same class. Conversely, the same word may be used by two teachers to describe two totally different classes, hence one teacher may use "greyhound" to describe the breed of dog while the other uses it to describe the C-2 navy aircraft. Similar problems occur when different teachers label the

data with different abstraction levels, so one generalizes over all dogs, while the other discriminates between a poodle, a Labrador and etc. Finally, teachers often do not agree on the exact demarcation of concepts, so a chair carved in stone may be labeled as a "chair" by one teacher, while the other describes it as "a rock". All these phenomena become increasingly pronounced as the number of classes is increased, thus their neglect essentially leads to a severe decrease in label purity and consequently in learning performance.

In this paper we study the cost of obtaining globally consistent labels, while focusing on a specific distributed labeling scenario, in which only some of the difficulties described above are present. To enforce the distributed nature of the problem, we assume that a large data set with n examples is to be labeled by a set of uncoordinated teachers, where each teacher agrees to label at most $l \ll n$ data points. While there is a one-to-one correspondence between the classes used by the different teachers, we assume that their labeling systems are entirely uncoordinated, so a class labeled as "duck" by one teacher may be labeled as a "goat" by another. In later stages of this paper, we relax this assumption, and consider a case in which partial consistency exists between the different teachers. Both scenarios are realistic in various problem domains. Consider for example a security system for which we have to label a large set of face images, including thousands of different people. Since teachers are not familiar with the persons to be labeled, the names they give to classes are entirely un-coordinated. The case of a partial consistency is exemplified in distributed labeling of flower images: the layman can easily distinguish between many different kinds of flowers but can name only a few.

The difficulties of "one-to-many" label correspondence between teachers and concept demarcation disagreements are not met by our current analysis, which focuses on the preliminary difficulties of distributed labeling. Another related scenario, to which our analysis can be extended relatively easily, is the case in which the initial data is labeled by uncoordinated teachers right from the start. Consider for example, the task of unifying images labeled in a site like Flickr[1] into a meaningful large training data set. Our suggested algorithms and analysis apply to this case with minor modifications.

1.1. Relevant literature

In the active learning framework (Cohn et al., 1990) and the experimental design framework (see e.g., (Atkinson & Donve, 1992)), the goal is to minimize the number of queries for labels (or experiments conducted) while learning a target concept. It has been shown (Freund et al., 1997) that a careful selection of queries can lead to an exponential reduction in the number of labels needed. This line of research is motivated by the costly and cumbersome process of obtaining labels for instances. We share this motivation but argue that the problem is not merely the quantity of labels but also the quality and the consistency of the labels that should be treated in the data collection process.

The problem of quality of labels, i.e., learning with noise, has been addressed extensively in the machine learning literature (see e.g., (Decator, 1995)). In this line of work it is assumed that the teacher does not always provide the true instance labels. The severity of noise ranges from adversarial noise, in which the teacher tries to prevent the learning process by providing inaccurate labels, to the more benign random classification noise. While the inconsistency between uncoordinated teachers can be regarded as some form of label noise, it has unique characteristics and its treatment is hence different from the other sources of noise mentioned. Specifically, as long as each teacher is noise-free and self-consistent, we are able to eliminate the noise completely and achieve certain labels.

The scenario of distributed labeling with uncoordinated teachers was considered in the "equivalence constraints" framework (Bar-Hillel et al., 2005). When learning with equivalence constraints, the learner is presented with pairs of instances and the annotation suggests whether they share the same class or not. The authors conjectured that as the number of classes increase, the labeling effort required to coordinate the labels from different teachers becomes prohibitive. We prove this conjecture in Theorem 3. Alternatively, equivalence constraints can be used as a direct supervision for the learning algorithm. Indeed, (Bar-Hillel & Weinshall, 2003) proved that a concept class is learnable with equivalence constraints if it is learnable from labels, so this alternative has some appeal.

1.2. The distributed labeling problem

In the distributed labeling task we have to reveal the labels of n instances $\{x_1, \ldots, x_n\}$. We assume that there exist "true" labels $y_1, \ldots, y_n$ (with $y_j = y(x_j)$) and the distributed labeling algorithm should return $\bar{y}_1, \ldots, \bar{y}_n$ such that $\bar{y}_i = \bar{y}_j$ if and only if $y_i = y_j$. We denote the number of classes by c, and assume that each teacher is willing to label only $l = c\alpha$ instances where $l, c \ll n$. Throughout this paper we assume that

[1] http://www.flickr.com/

the labels provided by teachers are consistent with the true labels in the sense that for any teacher t and any pair of instances x_i, x_j

$$[t(x_i) = t(x_j)] \Longleftrightarrow [y_i = y_j] \quad . \tag{1}$$

where $t(x)$ is the label given by teacher t to instance x. However, apart from section 4, we assume no inter-teacher consistency with respect to class names, i.e., teachers may disagree on the names of the different classes. To measure the competence of different algorithms for combining the labels of the different teachers we define the following:

Definition 1 *Denote by* $\sigma = \{x_i, y_i\}_{i=1}^{n}$ *an input sequence of* n *points with the labels* $y_i \in \{1, .., c\}$. *A distributed labeling algorithm* **alg** *is* $f(\alpha, \mathbf{alg})$ efficient *if*

$$f(\alpha, \mathbf{alg}) = \lim_{c \to \infty} \lim_{n \to \infty} \frac{n}{\sup_{\sigma} (\mathbf{labels}(\mathbf{alg}, \sigma, c\alpha))}$$

where $\mathbf{labels}(\mathbf{alg}, \sigma, l)$ *is the average (over the internal randomness of the algorithm) number of human-generated labels the algorithm* **alg** *uses to label the sequence* σ, *where each teacher is willing to label* l *examples.*

Clearly, if no structural assumptions are made on true labels then $f(\alpha, \mathbf{alg})$ is bounded by 1 from above. We denote by $f^*(\alpha)$ the optimal efficiency for a given α. I.e., $f^*(\alpha) = \sup_{\mathbf{alg}} f(\alpha, \mathbf{alg})$.

1.3. Main results

In section 2 we present several algorithms for solving the distributed labeling problem. The first algorithm presented is the *contract the connected components* (C^3) algorithm. We show that this simple algorithms has efficiency of $1 - {}^{(1-\exp(-\alpha))}/_{\alpha}$. We then improve this algorithm with the *representatives algorithm* and prove its efficiency to be better than the efficiency of the previous algorithm. In section 3 we present an upper bound on the achievable efficiency. We show that $f^*(\alpha) \leq \min({}^{2\alpha}/_{(1+\alpha)}, 1)$. In section 4 we study a relaxed version of the distributed labeling problem in which there exists some consistency between the different teachers. Thus, with some probability p two teachers will agree on the name of a given class. In this setting, we present a revised version of the C^3 algorithm and show its efficiency to be $1 - \frac{1-\exp(-\alpha)}{\alpha - \exp(-\alpha) + \exp(-\alpha(1-p))}$.

Algorithm 1 The *Contract* the *Connected Components* (C^3) algorithm

input: n unlabeled instances $x_1, \ldots, x_n$
output: a partition of $x_1, \ldots, x_n$ into classes according to the true labels

1. Let G be the edge-free graph whose vertexes are $x_1, \ldots, x_n$.
2. While G is not a clique
   (a) pick l random nodes $U = \{x_{i_1}, \ldots, x_{i_l}\}$ which are not a clique from G.
   (b) send U to a teacher and receive $y_{i_1}, \ldots, y_{i_l}$.
   (c) for every $1 \leq r < s \leq l$ do
      i. if $y_{i_r} = y_{i_s}$ then contract the vertices x_{i_r} and x_{i_s} in the graph G.
      ii. if $y_{i_r} \neq y_{i_s}$ then add the edge (x_{i_r}, x_{i_s}) to the graph G.
3. Mark each vertex in G with a unique number from $[1 \ldots c]$.
4. For every vertex in G, propagate its label to all the nodes that were contracted into this vertex.

2. Label-efficient algorithms

As described in 1.2, we assume in this section that the name each teacher assigns to a class is meaningless. Therefore, the best we can hope for is to break the n instances into c classes such that any pair of points share the same class label if and only if all teachers give these two points the same label. In this section we suggest two algorithms for this task. The bounds obtained for these algorithms are presented in Figure 1.

2.1. The *Contract* the *Connected Components* (C^3) algorithm

The first algorithm we consider is the *Contract* the *Connected Components* (C^3) algorithm presented in Algorithm 1. The idea behind this algorithm is to build a graph whose nodes are sets of equivalent instances. Whenever we find that two nodes share the same label, we contract them into a single node. On the other hand, whenever we find that two nodes do not share the same label, we generate an edge between them. The algorithm ends when the remaining graph is a clique. At this point, each of the nodes is assigned with a unique label. These labels propagate to all the points to be labeled, since each point is associated with a single node in the clique.

The correctness of the algorithm is straightforward due

to the self-consistency of the teachers. In Theorem 1 we show the label efficiency of the C^3 algorithm to be $1-(1-\exp(-\alpha))/\alpha$ where $\alpha = l/c$. The main idea behind the analysis is to study the expected number of contractions in each iteration.

Theorem 1 *The label efficiency of the C^3 algorithm is lower-bounded by*

$$1-\frac{1}{\alpha}(1-\exp(-\alpha)) \quad .$$

Before proving the theorem, we present a lemma in which the contraction rate associated with a single teacher is bounded.

Lemma 1 *Assume a teacher labels l random example ($l \to \infty$) from $c = l/\alpha$ different classes. The expected number of unique labels that the teacher will give to the l instances is at most l times $Q(\alpha)$ where*

$$Q(\alpha) = \frac{1}{\alpha}(1-\exp(-\alpha)) \quad .$$

Note that the number of unique labels is exactly the number of nodes that will be left after contracting the l instances.

Proof: Assume that the probability for seeing each of the classes is p_i. The result follows from the following:

$$\begin{aligned} & E[\text{number of unique labels}] \\ &= c - E[\text{number of labels not seen}] \\ &= c - \sum_i (1-p_i)^l \qquad (2) \\ &\le c - c\left(1-\frac{1}{c}\right)^l \\ &= c(1-\exp(-\alpha)) \qquad (3) \\ &= l\cdot\frac{1}{\alpha}(1-\exp(-\alpha)) \quad . \end{aligned}$$

The correctness of (3) follows since we are assuming that $l, c \to \infty$ while $\alpha = l/c$ is constant. □

Proof: *(of Theorem 1)* At each round of the C^3 algorithm, l elements are sent to be labeled by a teacher. From Lemma 1 we have that the number of remaining elements is on avarage at most $lQ(\alpha)$.

Therefore, the expected number of rounds the algorithm will make until finished is

$$\frac{n}{l\left(1-\frac{1}{\alpha}(1-\exp(-\alpha))\right)} \quad .$$

Note that the number in the denominator is the expected number of removed elements at each round. Thus, the number of labels used is

$$\frac{n}{\left(1-\frac{1}{\alpha}(1-\exp(-\alpha))\right)} \quad .$$

Plugging this number into the definition of label efficiency gives the desired result. □

2.2. The representatives algorithm

Each teacher provides us with two types of information sources. One is positive equivalence constraints, i.e., the knowledge that two instances share the same label. The other is negative equivalence constraints, i.e., the knowledge that two instances do not share the same label. While the C^3 algorithm is very effective in using positive equivalence constraints, it makes very little use of negative equivalence constraints. The *representatives algorithm* (Algorithm 2) tries to exploit this type of information as well. The main idea behind this algorithm is first to find all the points that belong to certain classes. Once we know that the remaining points do not belong to any of these classes, we are left with a problem with fewer instances and fewer potential classes and thus an "easier one".

In order to detect all the points belonging to a certain class we use *representatives*. A *representatives set* is a set of c instances $\{x_{i_1},..,x_{i_c}\}$such that for each class there is exactly one member (representative) of the class in the representatives set. Finding a representatives set is a simple task and can be done without affecting the overall efficiency, since its label complexity does not depend on n. Therefore, for the sake of simplicity we assume that the representatives set is given in advance. We further assume that we know the probability of each representative class. This information too can be easily estimated from data without jeopardizing efficiency.

β is the proportion of representatives in the l instances each teacher labels. Note that when $\beta = 0$, the representative algorithm is essentially the same as the C^3 algorithem. However, when $\beta > 0$, we use the fact that after all the points were compared against a certain representative, we are guaranteed to have found all the points with the same label as this representative, and thus we can eliminate this class.

Theorem 2 *The label efficiency of the representative algorithm is lower-bounded by*

$$\frac{(1-\beta)(1-q)^2}{1-q-\frac{q}{r}(1-q^r)}$$

where $r = \frac{c}{\beta l} = \frac{1}{\alpha\beta}$ is the number of sets in the

Algorithm 2 The Representatives Algorithm

Inputs:

- n unlabeled instances, $x_1, \ldots, x_n$
- a set $a_1, \ldots, a_c$ of representatives such that $a_i \in \{x_1, \ldots, x_n\}$
- a list of probabilities $p_1, \ldots, p_c$ such that p_i is the probability of seeing an instance from the class of a_i.

Outputs: a partition of the n points into c label classes

1. Reorder the representatives and the p_i's such that $p_1 \geq p_2 \geq \ldots \geq p_c$.
2. Let* $\beta \in (0,1)$
3. Partition the set of representatives into r sets $S_0, \ldots, S_{r-1}$ classes such that $S_i = \left\{a_{i\beta l+1}, \ldots, a_{(i+1)\beta l}\right\}$.
4. Let G be the edge free graph whose vertices are $x_1, \ldots, x_n$.
5. While G is not empty
   (a) For $i = 0 \ldots r-1$
      i. Partition the remaining points in the graph into sets of size $(1-\beta)\, l$.
      ii. For each subset of $(1-\beta)\, l$ points:
         A. send these points together with S_i to a teacher.
         B. contract the graph according to the labels returned by the teacher.
      iii. For every $a_j \in S_i$
         A. label a_j with the label j, and propogate this label.
         B. remove a_j from G.

* Choose β to optimize the bound in Theorem 2.

partition of the representatives into βl sets and[2] $q = Q(\alpha(1-\beta)) = \frac{1-\exp(-\alpha(1-\beta))}{\alpha(1-\beta)}$.

Proof: In each round of step 5a we break G into $|G| / (l(1-\beta))$ parts and thus use $|G| / (1-\beta)$ labels. Therefore, we need only to estimate the size of G after each round. Denote the number of vertices in G at the beginning of the round i by g_i. In order to bound g_i we should consider how it is affected by two ingredients: first the contraction which happen in the same fashion as it happens in the C^3 algorithm and the complete elimination of classes $1, .., i\beta l$.

We use Lemma 1 to analyze the contraction rate. Each teacher sees $l(1-\beta)$ instances which are not representers of some classes. These instances come from $c - i\beta l$ different classes and thus, from Lemma 1 the contraction rate is

$$Q\left(\frac{l(1-\beta)}{c-i\beta l}\right) = Q\left(\frac{\alpha(1-\beta)}{1-i\alpha\beta}\right) \quad .$$

Out of the remaining points, all the points which are being represented in S_i are eliminated. Due to the reordering of the p_is, these points are at least a fraction of $1/(r-i)$ of the remaining points. Thus

$$\begin{aligned} g_{i+1} &\leq g_i \frac{r-(i+1)}{r-i} Q\left(\frac{\alpha(1-\beta)}{1-i\alpha\beta}\right) \\ &= n\left(\prod_{j=0}^{i} \frac{r-(j+1)}{r-j}\right)\left(\prod_{j=0}^{i} Q\left(\frac{\alpha(1-\beta)}{1-j\alpha\beta}\right)\right) \\ &= n\left(1-\frac{i+1}{r}\right)\prod_{j=0}^{i} Q\left(\frac{\alpha(1-\beta)}{1-j\alpha\beta}\right) \quad . \end{aligned}$$

The number of labels used in all the rounds is therefore

$$\begin{aligned} &\sum_{i=0}^{r-1} \frac{g_i}{(1-\beta)} \leq \\ &\qquad \frac{n}{1-\beta}\sum_{i=0}^{r-1}\left(1-\frac{i}{r}\right)\prod_{k=0}^{i-1} Q\left(\frac{\alpha(1-\beta)}{1-k\alpha\beta}\right) \quad (4) \\ &\leq \frac{n}{1-\beta}\sum_{i=0}^{r-1}\left(1-\frac{i}{r}\right) Q(\alpha(1-\beta))^i \\ &= \frac{n\left(1-q-\frac{q}{r}(1-q^r)\right)}{(1-\beta)(1-q)^2} \end{aligned}$$

where (??) is due to the monotonicity of the Q function. Using the last expression in the efficiency definition completes the proof. □

The expression obtained in theorem 2 can be computed numerically for any value of α, β and so it can be used to optimize β for a given α. When the optimal β is used, the representers algorithm outperforms the C^3 algorithm as seen in Figure 1.

3. The optimal efficiency

In the previous section we studied the efficiency of several algorithms. In the current section we study the efficiency of the optimal algorithm. That is, we study

[2] The Q function is defined in Lemma 1.

the function

$$f^*(\alpha) = \sup_{\mathbf{alg}} f(\alpha, \mathbf{alg}) \quad .$$

We give an upper bound on $f^*(\alpha)$ showing that algorithms cannot have an efficiency greater than $\min(1, {}^{2\alpha}/_{(1+\alpha)})$. This bound asserts that the labeling problem is not trivial in the sense that it is not always possible to achieve efficiency 1. Moreover, the problem becomes hard in the limit of $\alpha \to 0$, as the efficiency drop linearly with α in this region. Comparing the bound shown here and the efficiency of the algorithms presented in previous sections, one can see that there is still a significant gap between the achieved and the (maybe) achievable.

Theorem 3 *Let $f^*(\alpha)$ be the best achievable efficiency for a given α then*

$$f^*(\alpha) \leq \min\left(1, \frac{2\alpha}{1+\alpha}\right) \quad .$$

Proof: Fix n and c and assume $l = \alpha c$. If $\alpha > 1$ then the required bound is trivial since efficiency cannot exceed 1. Therefore, we are only interested in the cases where $\alpha < 1$. Let **alg** be a distributed labeling algorithm. For each of the n instances we choose a class label uniformly and independently from the c possible labels. We analyze the expected number of teacher calls needed before the class assignments are found.

Fix an instance x, we first analyze the expected number of teacher calls (in which x participates) before it is first contracted with some other point. Assume that x has i edges in the graph G, i.e., there are i instances for which it is known that x does not share its label. If x' is a different point than x, the probability that they share the same label is at most ${}^1/_{(c-i)}$. To see this, note that for any legal label assignment to $G \setminus \{x\}$, there are at least $c - i$ uplifts of this assignment to G.

Let $P(i)$ be the probability that x is contracted at least once during its first i comparisons to other instances. We claim that $P(i) \leq {}^i/_c$ for all $1 \leq i \leq c$. Clearly, $P(0) = 0$. The proof is by induction. For $i = 1$, clearly the probability for contraction with the first point x is compared against is ${}^1/_c$. Note that

$$\begin{aligned} & P(i+1) \\ &= P(i) + (1 - P(i)) \Pr[\textbf{contract} \text{ at step } i+1] \\ &\leq P(i) + (1 - P(i)) \frac{1}{c-i} \\ &\leq \frac{i}{c}\left(1 - \frac{1}{c-i}\right) + \frac{1}{c-i} = \frac{i+1}{c} \quad . \end{aligned}$$

In the previous calculation, we assumed that x is compared to other points one at a time. However, the teachers label l instances at a time, thus whenever x is sent to a teacher, it is compared against $l - 1$ points. Note that an instance keeps being sent to teachers at least until it is first unified. Therefore, the number of teachers that will have to label x until its label is discovered, is at least the total number of teachers that will have to label x until it is unified at least once with another instance. From this we obtain the following lower bound for the expected number of teachers that see x:

$$\begin{aligned} & E[\text{number of teachers that see } x] \\ &= \sum_j \Pr[\text{number of teachers} \geq j] \\ &= \sum_j (1 - \Pr[\text{number of teachers} < j]) \\ &\geq \sum_{j=1}^{{}^{(c-1)}/_{(l-1)}} (1 - P((j-1)(l-1))) \\ &\geq \sum_{j=1}^{{}^{(c-1)}/_{(l-1)}} \left(1 - \frac{(j-1)(l-1)}{c}\right) \\ &= \frac{c-1}{l-1} - \frac{1}{2}\left(\frac{c-l}{l-1}\right)\left(\frac{c-1}{c}\right) \quad . \end{aligned}$$

The efficiency can be derived from this term

$$\begin{aligned} & f^*(\alpha) \\ &\leq 1/\lim_{c\to\infty}\left(\frac{c-1}{l-1} - \frac{1}{2}\left(\frac{c-l}{l-1}\right)\left(\frac{c-1}{c}\right)\right) \\ &= 1/\left(\frac{1}{\alpha} - \frac{1}{2}\left(\frac{1}{\alpha} - 1\right)\right) = \frac{2\alpha}{1+\alpha} \quad . \end{aligned}$$

□

4. Learning with name-consistent teachers

In previous sections we assumed that class names used by different teachers are totally uncoordinated, so naming conventions of one teacher are meaningless to the other. While this scenario may occur (like in the 'face labeling' task mentioned in the introduction), in most cases this assumption is too pessimistic. It is more reasonable to assume that some level of agreement regarding class names exist, though this agreement is partial and not perfect. In this section we assume that there exist $0 \leq p \leq 1$ such that with probability p over the choice of a random teacher t and class j, the teacher uses the true global class name j as the class label:

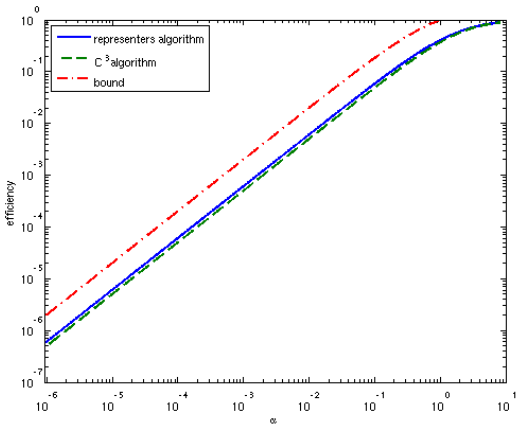

Figure 1. The efficiency (Y-axis) of the C^3 algorithm and the anchor algorithm are plotted together with the bound on the optimal efficiency (Theorem 3) for different values of α (X-axis).

$$\forall x \quad \Pr_t \left(t\left(x\right) = y\left(x\right) \right) \quad \geq \quad p \ . \tag{5}$$

We assume some sort of a probability measure over the teachers and the classes. If the pool of teachers is finite, it can be the uniform distribution, and otherwise we assume that whenever we need another teacher to label some instances, the teacher will be such that (5) is true. Notice that we also keep our previous assumption that all the teachers are class consistent in the sense of (1).

When $p = 1$ the assumption (5) means that all the teachers use the same global naming system , i.e. $t(x_j) = y_j$ for all t, j. In this case the labeling problem is trivial, and it is easy to obtain label efficiency of 1 simply by splitting the instances between different teachers. On the other hand, when p is very small, there is no name consistency and the situation boils down to the scenario studied in Section 2. Therefore, we will now focus on studying name consistency in the general case when $p \in (0, 1)$.

The algorithm we present to address this situation is the *Consistently Contract* the *Connected Components* (C^4) (Algorithm 3). The difference between the C^4 algorithm and the C^3 algorithm is that the C^4 algorithm sends teachers instances that were previously given the same label by some other teachers.

The C^4 algorithm differs from the C^3 algorithm in using the labels for selecting better candidates for sending to the same teacher. However, note that we still declare the equivalence of two instances only when a single teacher labels both with the same label. Therefore, due to the class consistency (1) the correctness of the algorithm is guaranteed. We now turn to proving its efficiency.

Algorithm 3 The Consistently Contract the Connected Components (C^4) algorithm

Input: n unlabeled instances $x_1, \ldots, x_n$
Output: a partition of $x_1, \ldots, x_n$ into classes according to the true labels

1. Let G be the edge free graph whose vertices are $x_1, \ldots, x_n$.
2. Label each vertex with 0.
3. While G is not a clique
   (a) pick l random nodes $U = \{x_{i_1}, \ldots, x_{i_l}\}$ from G such that all these nodes have the same label.
   (b) send U to a teacher and receive $y_{i_1}, \ldots, y_{i_l}$.
   (c) for every $1 \leq r \leq l$, label x_{i_r} with the label y_{i_r}.
   (d) for every $1 \leq r < s \leq l$ do
      i. if $y_{i_r} = y_{i_s}$ then contract the vertices x_{i_r} and x_{i_s} in the graph G.
      ii. if $y_{i_r} \neq y_{i_s}$ then add the edge (x_{i_r}, x_{i_s}) to the graph G.
4. Mark each vertex in G with a unique number.
5. For every vertex in G propagate its label to all the nodes that were contracted into this vertex.

Theorem 4 *The label efficiency of the C^4 algorithm is lower bounded by*

$$1 - \frac{1 - \exp\left(-\alpha\right)}{\alpha - \exp\left(-\alpha\right) + \exp\left(-\alpha\left(1-p\right)\right)}$$

Proof: Following the proof of the efficiency of the C^3 algorithm, we compute the rate in which the size of G reduces. However, we need to consider two settings. The first applies to teachers that label points for the first time. The second case to consider is teachers who label points that were previously labeled by some other teacher. While these cases may be interleaved in time according to algorithm C^4, w.l.o.g. we may analyze them as if they occur in two consecutive phases.

Following Lemma 1, teachers who label points that were not previously labeled will leave for further process $lQ\left(\alpha\right)$ points out of every l labeled points. Thus the first phase of labeling will require n labels and will leave $nQ\left(\alpha\right)$ points in the graph G.

In the second phase, each teacher is fed with points that received the same label by different teachers. Due to the name consistency (5) out of l points that a teacher labeled we expect pl of them to have the same label due to the name consistency. The other points are subject to contraction. From Lemma 1 and the above argument we expect that from every l points only $1+(1-p)\,lQ\left(\alpha\left(1-p\right)\right)$ will remain. The number of labels used by teachers labeling previously labeled points is

$$\frac{nQ\left(\alpha\right)}{1-\left(1-p\right)Q\left(\alpha\left(1-p\right)\right)-\frac{1}{l}}$$

Thus, the overall number of labels used is

$$n\left(\frac{Q\left(\alpha\right)}{1-\left(1-p\right)Q\left(\alpha\left(1-p\right)\right)-\frac{1}{l}}+1\right)$$

which leads to the efficiency of

$$\lim_{l\to\infty}\frac{1-\left(1-p\right)Q\left(\alpha\left(1-p\right)\right)-\frac{1}{l}}{Q\left(\alpha\right)+1-\left(1-p\right)Q\left(\alpha\left(1-p\right)\right)-\frac{1}{l}}=$$
$$1-\frac{1-\exp\left(-\alpha\right)}{\alpha-\exp\left(-\alpha\right)+\exp\left(-\alpha\left(1-p\right)\right)}$$

□

One can easily verify, that if $p=0$ the label efficiency of the C^4 algorithm is identical to that of the C^3 algorithm. However, the difference between the C^3 algorithm and C^4 algorithm is profound when $p\to 1$ and $\alpha\to 0$. In this setting, the C^3 algorithm has efficiency of $(\alpha/2)+o\left(\alpha\right)$ while the C^4 algorithm is $(1/2)-o\left(1\right)$ efficient.

Note that despite the remarkable improvment, when $p=1$ there exists complete name consistency and thus it is trivially possible to achieve the perfect efficiency of 1. However, it is not clear if it is possible to get efficiency close to 1 if p is slightly less than 1. This remains as an open problem.

5. Conclusions and further research

In this work we have studied the problem of generating consistent labels for a large data set given that the labels are provided by restricted teachers. We have focused on the problems arising when the labels used by different teachers are un-coordinated, but nevertheless a one-to-one (unknown) correspondence exists between their labeling systems. In this framework, we provided several algorithms and analyzed their efficiency. We also presented an upper bound which shows that the problem is non-trivial, and becomes hard as the number of classes grows. In the limit $\alpha\to 0$ we characterize the achievable efficiency to be in the range[3] $[(2/3)\,\alpha, 2\alpha]$, however the exact value remains as an open problem.

We believe that the process of collecting data for large scale learning deserves much attention. One interesting extension of this work is to the case where the symmetry between teachers is broken, either by considering different noise levels to their labels, or more generally, by also allowing the noise level to change between the different classes. In such scenarios, a 'teacher selection' problem arises as the identity of the teacher can be very informative. One example is the problem of "provost-selection" in which most of the teachers are useless novices in some domain-specific issues and thus it is essential to first find the experts ("provosts") and use only the labels they provide. A related problem arises when all teachers are useful, but they differ in their discrimination resolutions, so one teacher may say that an image contains a bird while the other may describe the exact bird species. Such problems are left for further research.

References

Atkinson, A. C., & Donve, A. N. (1992). *optimum experiment designs*. Oxford University Press.

Bar-Hillel, A., Hertz, T., Shental, N., & Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Reseach (JMLR), 6(Jun)*, 937–965.

Bar-Hillel, A., & Weinshall, D. (2003). Learning with equivalence constraints, and the relation to multiclass classification. *Conference on Learning Theory (COLT)*.

Cohn, D., Atlas, L., & Ladner, R. (1990). Training connectionist networks with queries and selective sampling. *Advanced in Neural Information Processing Systems 2*.

Decator, S. E. (1995). *Efficient learning from faulty data*. Doctoral dissertation, Harvard University.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. `http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html`.

Freund, Y., Seung, H., Shamir, E., & Tishby, N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning, 28*, 133–168.

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset* (Technical Report 7694). California Institute of Technology.

Russell, B., Torralba, A., Murphy, K., & Freeman, W. (2005). Labelme: a database and web-based tool for image annotation. mit ai lab memo aim-2005-025.

von Ahn, L. (2006). Games with a purpose. *IEEE Computer, 39*, 92–94.

[3] The representers algorithm achieves efficiency of $(2/3)\,\alpha$ with $\beta=1/3$ and $\alpha\to 0$. To see this, plug these values in (4).