2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Learning in Presence of Ontology Mapping Errors

Neeraj Koul
Iowa State University,Ames, IA, USA
Email: neeraj@cs.iastate.edu

Vasant Honavar
Iowa State University, Ames, IA, USA
Email: honavar@cs.iastate.edu

***Abstract*—The widespread use of ontologies to associate semantics with data has resulted in a growing interest in the problem of learning predictive models from data sources that use different ontologies to model the same underlying domain (world of interest). Learning from such *semantically disparate* data sources involves the use of a mapping to resolve semantic disparity among the ontologies used. Often, in practice, the mapping used to resolve the disparity may contain errors and as such the learning algorithms used in such a setting must be robust in presence of mapping errors. We reduce the problem of learning from semantically disparate data sources in the presence of mapping errors to a variant of the problem of learning in the presence of *nasty* classification noise. This reduction allows us to transfer theoretical results and algorithms from the latter to the former.**

***Keywords*-semantically disparate data sources ; ontology mapping errors; nasty classification noise; PAC;**

I. Introduction

Recent advances in high throughput data acquisition technologies in many applications have resulted in a proliferation of autonomous and distributed data sources. Different data sources often use disparate vocabularies (e.g., M.S. student versus Masters student), units (e.g., temperature measured in degrees Centigrade versus Fahrenheit), and levels of detail (e.g. graduate student, student) to describe the objects of interest in the world being modeled. In such a setting, different data sources represent different conceptual models of the same underlying world. In the semantic web vision this typically translates to each data source assuming a particular ontology to model objects, properties and relationships in the world of interest. Hence, learning from such data sources requires reconciling the semantic differences between the learner's conceptual model of the world (i.e., learner's ontology) and the models of the world associated with the the disparate data sources (i.e., the data source ontologies). This is achieved through a data integration step [1] [2] that presents to the learning algorithm, a single *view* of the different data sources. Data integration involves mapping the terms in the data source ontologies to the learner's ontology (See [3] for a survey). However, this mapping process is often error prone. Errors in mappings can be due to human error, errors in the automated mapping algorithm used, or by lack of exact correspondences between terms in a source ontology and the target ontology. Hence, it is of interest to characterize the effect of mapping errors on the accuracy of the predictive models (e.g., classifiers) learned in such a setting.

Against this background, this paper reduces the problem of learning from semantically disparate data sources in the presence of mapping errors to a variant of the problem of learning in the presence of *nasty* classification noise in a PAC-like framework (see [4], [5] for background on PAC learning). This reduction proves to be very useful in practice as techniques to deal with noise have been well studied in literature and can be applied to the setting of learning in presence of mapping errors.

II. Learning from Semantically Disparate Data

We introduce the notion of a *k-Delegating Oracle* to model learning from multiple data sources. We then extend the model to a *mapping aware k-Delegating Oracle* to model learning from semantically disparate data sources.

A. k-*Delegating Oracle*

Let $\mathcal{X}$ be an instance space, $\mathcal{D}$ a probability distribution over $\mathcal{X}$, $\mathcal{F}$ a function space and $f : \mathcal{X} \longrightarrow \{0, 1\}$ the target function to be learned ($f \in \mathcal{F}$) (assume binary classification for simplicity). An oracle $EX(f, \mathcal{X}, \mathcal{D})$ is a procedure that returns a labeled example $\langle x, f(x)\rangle$ where x is drawn from $\mathcal{X}$ according to $\mathcal{D}$. We use the notation $Pr_{x\in\mathcal{D}}[x]$ to indicate the probability of drawing an instance x from $\mathcal{X}$ according to the distribution $\mathcal{D}$. The classical model of supervised learning, consisting of a learner L with access to an Oracle $EX(f, \mathcal{X}, \mathcal{D})$, is not expressive enough to model learning from multiple data sources. Consequently we introduce the notion of a *k-Delegating Oracle* to model learning from multiple data sources.

A *k-delegating oracle* $kEX(f, \mathcal{X}, \mathcal{D})$ invokes subordinate oracles $EX^1(f, \mathcal{X}, \mathcal{D}_1), \ldots EX^k(f, \mathcal{X}, \mathcal{D}_k)$ with probabilities $p_1 \ldots p_k$ respectively. The i^{th} oracle $EX^i(f, \mathcal{X}, \mathcal{D}_i)$ when invoked returns an example of the form $\langle x, f(x)\rangle$ where x is drawn from $\mathcal{X}$ according to $\mathcal{D}_i$. The distribution $\mathcal{D}$ of the *k*-delegating oracle is $Pr_{x\in\mathcal{D}}[x] = \sum_{i=1}^{k} p_i \times Pr_{x\in\mathcal{D}_i}[x]$

B. *Mapping Aware* k-*Delegating Oracle*

Let $\mathcal{X}_{s^1}, \mathcal{X}_{s^2} \ldots \mathcal{X}_{s^k}$ be k instances spaces; Let $\mathcal{D}_1, \mathcal{D}_2 \ldots \mathcal{D}_k$ be probability distributions over $\mathcal{X}_{s^1}, \mathcal{X}_{s^2} \ldots \mathcal{X}_{s^k}$ respectively and $\mathcal{F}^1, \mathcal{F}^2 \ldots \mathcal{F}^k$ be k

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.138

functions spaces defined over the corresponding instance spaces where each function in $\mathcal{F}^i$ labels instances in $\mathcal{X}_{s^i}$ with a label in the set C_i.

A *mapping aware k-delegating oracle* has access to a mapping set $M = \{m_1, m_2 \ldots m_k\}$ where $m_i = \{m_i^x, m_i^c\}$; $m_i^x : \mathcal{X}_{s^i} \longrightarrow \mathcal{X}$ is an *attribute mapping function*; and $m_i^c : C_i \longrightarrow C$ is a *class mapping function* where $C_i = Range(f_i)$ and $C = Range(f)$. It invokes subordinate oracles $EX^1(f_1, \mathcal{X}_{s^1}, \mathcal{D}_1) \ldots EX^k(f_k, \mathcal{X}_{s^k}, \mathcal{D}_k)$ where the i^{th} subordinate oracle $EX^i(f_i, \mathcal{X}_{s^i}, \mathcal{D}_i)$ returns examples of the form $\langle x_{s^i}, f_i(x_{s^i})\rangle$ where x_{s^i} is drawn from X_{s^i} according to $\mathcal{D}_i$ and $f_i \in \mathcal{F}^i$. It uses the mapping m_i to convert an instance $\langle x_{s^i}, f_i(x_{s^i})\rangle$ received form the i^{th} subordinate oracle to $\langle m_i^x(x_{s^i}), m_i^c(f_i(x_{s^i}))\rangle$ before passing it to the learner. We assume the mappings m_i are computable and satisfy the following conditions: $\forall x_{s^i} \in \mathcal{X}_{s^i}, m_i^x(x_{s^i}) \in \mathcal{X}$; $\forall l \in C_i, m_i^c(l) \in C$ and whenever $x \in \mathcal{X}_{s^i}, \mathcal{X}_{s^j}$, $m_i^x(x) = m_j^x(x)$. These conditions ensure that the examples returned by the mapping aware *k*-delegating oracle are of the form $\langle x, l(x)\rangle$ where $x \in \mathcal{X}$ and $l(x) \in C$.

Ideally the mappings should ensure that the examples returned to the learner are labeled according to the target function f. However, in practice mappings may have errors and consequently the instances may be labeled according to ϕ which may be different from f. We denote the mapping aware *k*-delegating oracle by $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ where ϕ is the labeling function.

From a learner L's point of view (that uses the mapping aware k-delegating oracle) we need to describe, in addition to the labeling function ϕ, the distribution $\mathcal{D}$ (over $\mathcal{X}$) with which the instances are sampled. Let $Y_{s^i}^x$ be a set that consists of all the elements in $\mathcal{X}_{s^i}$ that are mapped to an element $x \in \mathcal{X}$ using the mapping m_i^x. Then the distribution $\mathcal{D}$ over $\mathcal{X}$ is given by

$$Pr_{x\in\mathcal{D}}[x] = \sum_{i=1}^{k} \sum_{y \in Y_{s^i}^x} p_i \times Pr_{y\in\mathcal{D}_i}[y] \qquad (1)$$

Note that the sampling distribution $\mathcal{D}$ now depends on mappings $m_1^x \ldots m_k^x$ (because of dependence on $Y_{s^i}^x$).

III. Learning in the Presence of Mapping Errors

We now proceed to describe (formally) what it means for a mapping to be correct (and correspondingly to have errors) and establish an equivalence between learning in the presence of mapping errors and learning from noisy data.

A. Mapping Errors

The sets of class labels $C_1 \ldots C_k$ as well C partition the corresponding instance spaces $\mathcal{X}_{s_1} \ldots \mathcal{X}_{s_k}$ and $\mathcal{X}$ respectively. Each cell in a partition corresponds to a set of instances that share the same class label. The mapping m_i^c establishes a correspondence between the cells of the partition of $\mathcal{X}_{s_i}$ and those of the partition of $\mathcal{X}$. We define

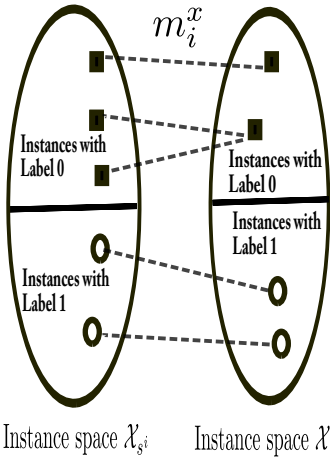

Figure 1. An example of a correct mapping.

errors in mappings relative to a reference set of mappings $m_{1,expert}^c(l) \ldots m_{k,expert}^c(l)$ (e.g., provided by an expert).

***Definition** 1 (Correct Class Mapping):* A class mapping m_i^c is said to be correct if $\forall l \in C_i \; m_i^c(l) = m_{i,expert}^c(l)$.

***Definition** 2 (Correct Attribute Mapping):* An attribute mapping m_i^x is said to be correct whenever $\forall x \in \mathcal{X}_{s^i}$, $f_i(x) = l$ and $m_{i,expert}^c(l) = l_1 \longrightarrow f(m_i^x(x)) = l_1$

***Definition** 3 (Correct Mapping Set):* A mapping set $M = \{m_1, m_2 \ldots m_k\}$ is said to be correct if $\forall i \in \{1, 2, \ldots, k\}$ m_i^x and m_i^c are correct.

Observation 1: Given a correct mappings set M, for each labeled example of the form $\langle x, \phi(x)\rangle \in \mathcal{X} \times C$ provided by $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$, it must be the case that $\phi(x) = f(x)$ where f is the target function.

Observation 1 shows that when the mappings have no errors the instances passed to the learner are labeled according to the target function f. In the rest of the paper, we assume that a correct class label mapping is available (say from a domain expert) and all mapping errors are attribute mapping errors. An example of an correct mapping and an incorrect mapping is shown in Figure 1. and Figure 2. respectively.

B. Mapping Errors as Noise

We show that the mapping errors manifest themselves as classification noise in the labeled examples provided to the learner. Let $Pr_{x\in\mathcal{D}}[e = \langle x, f(x)\rangle]$ denote the probability that a labeled example $e = \langle x, f(x)\rangle$ is obtained by a single call to the oracle $EX(f, \mathcal{X}, \mathcal{D})$.

***Definition** 4 (Equivalent Oracles):* The oracles $EX1(f_1, \mathcal{X}, \mathcal{D}_1)$ and $EX2(f_2, \mathcal{X}, \mathcal{D}_2)$ are said to be equivalent whenever $\forall e \in \mathcal{X} \times Range(f_1) \cup Range(f_2), Pr_{x\in\mathcal{D}_1}[e = \langle x, f_1(x)\rangle] = Pr_{x\in\mathcal{D}_2}[e = \langle x, f_2(x)\rangle]$

The following observation follows directly from Observation 1 and the definition of equivalent oracles.

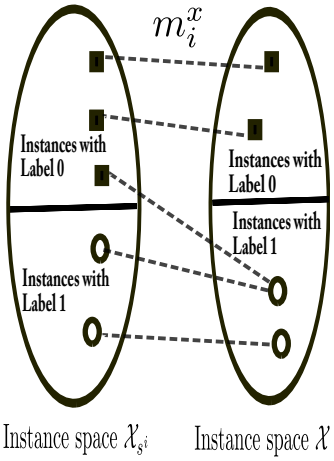

Figure 2. An example of a mapping with errors.

Observation 2: A k-delegating oracle $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ is equivalent to a classical oracle $EX(f, \mathcal{X}, \mathcal{D})$ whenever the mapping set M is correct with respect to target function f.

***Definition** 5 (Noisy Oracle):* Let $\eta_x : \mathcal{X} \mapsto [0, 1]$ be an *instance dependent classification noise rate*. A noisy oracle $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D}_{eq})$ operates as follows: It calls a classical oracle $EX(f, \mathcal{X}, \mathcal{D}_{eq})$ to obtain a labeled example $\langle x, f(x)\rangle$ and returns to the learner the example $\langle x, f(x)\rangle$ with a probability $1 - \eta_x$ and $\langle x, 1 - f(x)\rangle$ with probability η_x.

Given a k-delegating oracle $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$, let $\beta(x)$ be the probability that an instance x obtained by a single call to $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ has the label $\phi(x)$ which is different from $f(x)$. Let $\gamma(x)$ be the probability that an instance x obtained by a single call to $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ has the label $\phi(x)$ which is same as $f(x)$.

***Theorem** 1:* A k-delegating oracle $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ is equivalent to a noisy oracle $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D}_{eq})$ when the distributions $\mathcal{D}$ and $\mathcal{D}_{eq}$ are identical and $\eta_x = \frac{\beta(x)}{\beta(x)+\gamma(x)}$

Proof: From (1), the distribution $\mathcal{D}$ over $\mathcal{X}$ of the given k-delegating oracle is

$$Pr_{x\in\mathcal{D}}[x] = \sum_{i=1}^{k} \sum_{y\in Y_{s^i}^x} p_i \times Pr_{y\in\mathcal{D}_i}[y]$$

We define $\alpha_i(x) = \sum_{y\in Y_{s^i}^x} Pr_{y\in\mathcal{D}_i}[y]$, then

$$Pr_{x\in\mathcal{D}}[x] = \sum_{i=1}^{k} p_i \times \alpha_i(x)$$

Now $\alpha_i(x)$ can be seen as the weight (sum of probabilities) of instances drawn from X_s^i according to $\mathcal{D}_i$ that is mapped to $x \in \mathcal{X}$. In presence of mapping errors let the set $Y_{s^i}^x = A_{s^i}^x \cup B_{s^i}^x$ where $A_{s^i}^x$ is subset of instances in $Y_{s^i}^x$ that are correctly mapped to $x \in \mathcal{X}$ while $B_{s^i}^x$ is the subset of instances in $Y_{s^i}^x$ that get mapped to $x \in \mathcal{X}$ due to mapping errors. Let $\gamma_i(x) = \sum_{y\in A_{s^i}^x} Pr_{y\in\mathcal{X}_{s^i},\mathcal{D}_i}[y]$ and $\beta_i(x) = \sum_{y\in B_{s^i}^x} Pr_{y\in\mathcal{X}_{s^i},\mathcal{D}_i}[y]$. Note that $\beta_i(x)$ and $\gamma_i(x)$ (respectively) are the weights of instances drawn from X_s^i according to $\mathcal{D}_i$ that are incorrectly and correctly mapped to $x \in \mathcal{X}$. Recall that $x \in X_s^i$ is correctly mapped using m_i^x if the following holds

$$f_i(x) = l_1 \; and \; m_{i,expert}^c(l_1) = l \longrightarrow f(m_i^x(x)) = l.$$

It follows that

$$\alpha_i(x) = \beta_i(x) + \gamma_i(x)$$

In addition $\gamma(x) = \sum_{i=1}^{k} p_i \times \gamma_i(x)$ and $\beta(x) = \sum_{i=1}^{k} p_i \times \beta_i(x)$. Note that $\beta(x)$ is the probability that given the instance x is drawn (from $\mathcal{X}$ according to $\mathcal{D}$), it is labeled incorrectly . Similarly $\gamma(x)$ is the probability that given the instance x is drawn (again from $\mathcal{X}$ according to $\mathcal{D}$), it is labeled correctly. Hence

$$Pr_{x\in\mathcal{D}}[x] = \gamma(x) + \beta(x)$$

To avoid cluttering the notation we will abbreviate $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ and $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D}_{eq})$ by kEX and $EX1_{\eta_x}$ respectively when the parameters are obvious from the context. Consider a labeled example $e = \langle x, l(x)\rangle \in E = \mathcal{X} \times \{0, 1\}$ where $l(x)$ is the label associated with x. The labeled example $e = \langle x, l(x)\rangle$ can be sampled either from kEX or $EX1_{\eta_x}$ and since the class labels are binary $l(x)$ is either $f(x)$ of $1 - f(x)$.

case : $l(x) = f(x)$

$$Pr_{x\in\mathcal{D}_{eq}}[e = \langle x, f(x)\rangle] = (1 - \eta_x) Pr_{x\in\mathcal{D}_{eq}}[x]$$
$$Pr_{x\in\mathcal{D}}[e = \langle x, f(x)\rangle] = \gamma(x)$$

case: $l(x) = 1 - f(x)$

$$Pr_{x\in\mathcal{D}_{eq}}[e = \langle x, 1 - f(x)\rangle] = \eta_x Pr_{x\in\mathcal{D}_{eq}}[x]$$
$$Pr_{x\in\mathcal{D}}[e = \langle x, f(x)\rangle] = \beta(x)$$

When the noisy oracle $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D}_{eq})$ is such that

$$Pr_{x\in\mathcal{D}_{eq}}[x] = Pr_{x\in\mathcal{D}}[x] = \gamma(x) + \beta(x) = \sum_{i=1}^{k} p_i \times \alpha_i(x)$$

and

$$\eta_x = \frac{\beta(x)}{\beta(x) + \gamma(x)}$$

it follows that for either case

$$Pr_{x\in\mathcal{D}_{eq}}[e = \langle x, f(x)\rangle] = Pr_{x\in\mathcal{D}}[e = \langle x, f(x)\rangle] \qquad (2)$$

This establishes the equivalence of oracles $EX1_{\eta_x}$ and kEX.

The theorem shows that the effect of the mapping errors in the k-delegating oracle kEX can be simulated by the noise function η_x associated with $EX1_{\eta_x}$.

C. Mapping Errors as Nasty Noise

We now argue that the noise model η_x associated with $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D})$ can be simulated by the nasty classification noise [6] model which in turn can be simulated by the nasty sample noise model [6].

Definition 6: **(Instance Dependent Classification Noise(IDCN) Oracle)**: An Instance Dependent Classification Noise Oracle, denoted by $IDCN(m, \eta_x, f, \mathcal{X}, \mathcal{D})$, is one where an intermediary obtains a dataset D^m of m i.i.d examples by making m calls to a noisy oracle $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D}_{eq})$. The resulting dataset is then provided to the learner.

Definition 7: **(k^m-delegating Oracle)**: A k^m-delegating Oracle, denoted by $kEX^m(\phi, \mathcal{X}, \mathcal{D}, M)$, is one where an intermediary obtains a dataset D^m of m i.i.d examples by making m calls to a *k*-delegating oracle $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$. The resulting dataset is then provided to the learner.

Definition 8: **(Nasty Sample Noise (NSN) Oracle (adapted from [6]))**: A Nasty Sample Noise Oracle, denoted by $NSN(m, \eta, f, \mathcal{X}, \mathcal{D})$, is one where an adversary obtains a dataset D^m of m i.i.d examples by making m calls to a classical oracle $EX(f, \mathcal{X}, \mathcal{D})$. The adversary then picks n out of m instances of its choosing from D^m (where n is distributed according to a binomial distribution with parameters m and nasty noise rate η) and replaces them with any examples of its choice from $\mathcal{X} \times Range(f)$. The resulting dataset is then provided to the learner.

Definition 9: **(Nasty Classification Noise (NCN) Oracle (adapted from [6]))**: A Nasty Classification Noise Oracle, denoted by $NCN(m, \eta, f, \mathcal{X}, \mathcal{D})$, is one where an adversary obtains a dataset D^m of m i.i.d examples by making m calls to a classical oracle $EX(f, \mathcal{X}, \mathcal{D})$. The adversary then picks n out of m instances of its choosing from D^m (where n is distributed according to a binomial distribution with parameters m and nasty noise rate η) and flips their class labels. The resulting dataset is then provided to the learner.

Nasty Classification Noise (NCN) is a *weaker* case of NSN where the adversary is constrained such that it can modify only the class labels of n instances selected from D^m in a manner identical to that in the case of NSN

Consider a dataset obtained from $IDCN(m, \eta_x, f, \mathcal{X}, \mathcal{D})$. Let $\lambda = \sum_{x \in \mathcal{X}} \eta_x \times Pr_{x \in \mathcal{D}}[x]$. The value λ represents the probability that a random example in the dataset obtained by a single call to $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D})$ is mislabeled. The number of examples in the dataset obtained from $IDCN(m, \eta_x, f, \mathcal{X}, \mathcal{D})$ that have incorrect labels with respect to f can be viewed as number of successes in a sequence of m independent binary experiments each with a success probability λ. In the case of $NCN(m, \eta, f, \mathcal{X}, \mathcal{D})$, if we choose $\eta = \lambda = \sum_{x \in \mathcal{X}} \eta_x \times Pr_{x \in \mathcal{D}}[x]$, it follows that the number of incorrectly labeled examples in the dataset can also be viewed as number of successes in a sequence of m independent binary experiments each with a success probability λ. However, the n examples that are mislabeled in the dataset obtained from $IDCN(m, \eta_x, f, \mathcal{X}, \mathcal{D})$ are determined by function η_x whereas in the case of a dataset obtained from $NCN(m, \eta, f, \mathcal{X}, \mathcal{D})$ *any* n of the m instances can be mislabeled (For example the label of an instance x for which $\eta_x = 0$ will never be mislabeled in a dataset obtained from $IDCN(m, \eta_x, f, \mathcal{X}, \mathcal{D})$ whereas it is possible that the same instance can be mislabeled in a dataset obtained from $NCN(m, \eta, f, \mathcal{X}, \mathcal{D})$). The preceding argument leads to the following observation:

Observation 3: The IDCN model can be simulated by the NCN model and hence also by the NSN model.

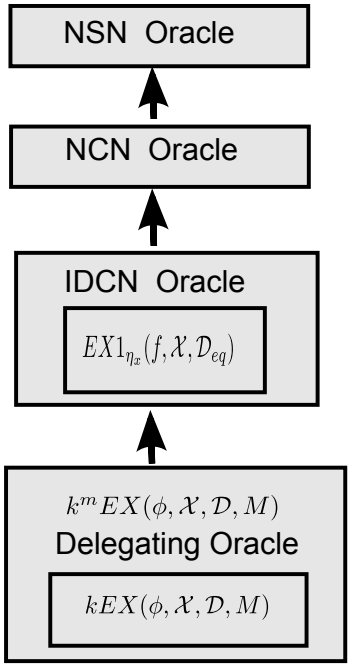

Figure 3. A schematic representation of hierarchy between types of oracles (the arrows denote can be simulated by).

The IDCN oracle uses a noisy oracle $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D}_{eq})$ while the $kEX^m(\phi, \mathcal{X}, \mathcal{D}, M)$ oracle uses a *k*-delegating oracle $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$. However, Theorem 1 states that every *k*-delegating Oracle has an *equivalent* Noisy Oracle. This leads to the following observation:

Observation 4: The $kEX^m(\phi, \mathcal{X}, \mathcal{D}, M)$ Oracle can be simulated by the IDCN model.

Observation 3 and observation 4 results in a hierarchy of Oracles and is depicted in Figure 3. It follows, in a straightforward way, from the hierarchy of oracles (see Figure 3), that learning in presence of ontology mapping errors can be seen as a weaker case of learning with nasty classification noise. As a result a learner can apply the same techniques to deal with mapping errors that it applies to deal with nasty classification noise. This result proves to be very useful in practice, since techniques to deal with

noise have been studied extensively in literature and can be ported in a straightforward way to the setting of learning in presence of mapping errors. For example, similar to learning from noisy data, learning in presence of mapping errors is prone to overfitting and may be addressed by pruning [7] [8] [9]. Similarly, on the lines of eliminating class noise (see approaches in [10] [11] [12]) filtering instances with mapping errors may be used to improve the performance of the classifiers learned in the presence of mapping errors. In addition, insights from noisy learning can be borrowed to to the setting of learning in presence of mapping errors, e.g. the classifier AdaBoost whose performance is known to degrade in presence of classification errors (see [13]), is not a good choice to learn in the setting of mapping errors (a good choice may be Robust Alternating AdaBoost [14], an noise-tolerant version of AdaBoost and hence also tolerant to mapping errors).

D. Learning in the Presence of Mapping Errors

We now proceed to present some theoretical results for learning in presence of mapping errors in a PAC like setting.

***Definition** 10 (PAC Learnability (from [5])):* A class F of boolean functions is PAC-learnable using hypothesis class $\mathcal{H}$ in polynomial time if there exists an algorithm that, for any $f \in \mathcal{H}$, any $0 < \epsilon < 1/2$, $0 < \delta < 1$ and any distribution $\mathcal{D}$ on $\mathcal{X}$, when given access to the PAC oracle, runs in time polynomial in $log|\mathcal{X}|, 1/\epsilon, 1/\delta$ and with probability at least $1-\delta$ outputs a function $h \in \mathcal{H}$ for which $Pr_{x \in \mathcal{D}}[h(x) \neq f(x)] \leq \epsilon$.

***Definition** 11 (Mapping Error Rate):* The mapping error rate of a *k*-delegating oracle $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ is defined as the probability that an example $\langle x, \phi(x) \rangle$ obtained by making a call to $kEX(\phi, \mathcal{X}, \mathcal{D}, M)$ has a label that is different from that assigned by the target function f.

Observation 5: The mapping error rate of a *k*-delegating oracle $kEX(\phi, \mathcal{X}, \mathcal{D}, M) = \sum_{x \in \mathcal{X}} \beta(x) \times Pr_{x \in \mathcal{D}}[x]$.

PAC learning is information theoretically impossible in the case when the probability that a randomly drawn example from an oracle has an incorrect label ≥ 0.5. Hence PAC learning is not possible in the case of Noisy oracle $EX1_{\eta_x}(f, \mathcal{X}, \mathcal{D}_{eq})$ when $\forall x, \eta_x > 0.5$ or in the case of $NCN(m, \eta, f, \mathcal{X}, \mathcal{D})$ when $\eta \geq 0.5$. Correspondingly, PAC learning is also not possible when the mapping error rate $\beta \geq 0.5$. This result provides an upper bound on the amount of mapping errors that can be tolerated.

Consider NastyConsistent, a PAC learning algorithm under the NSN model [6].

Algorithm NastyConsistent

Input: certainty parameter $\delta > 0$, the nasty error rate $\eta < \frac{1}{2}$ and required accuracy $\epsilon = 2\eta + \Delta$.

begin

1) Request a sample $S = \{\langle x, l(x) \rangle\}$ of size $m > \frac{c}{\Delta^2}(d + log2/\delta)$ from the NSN oracle.
2) Output any $h \in \mathcal{F}$ such that $|\{x \in S : h(x) \neq l(x)\}| \leq m(\eta + \Delta/4)$ (if no such h exists, output any $h \in \mathcal{F}$).

end

***Theorem** 2:* (Restatement of theorem 4 in [6]) Let $\mathcal{C}$ be any class of VC-dimension d (See [15] for background on VC-dimension). Then, there exists a choice of the constant c for which NastyConsistent is a PAC learning algorithm under nasty sample noise of rate η.

Proof: See proof of Theorem 4 in [6]. ■

Consider the following variant of the NastyConsistent algorithm which uses $kEX^m(\phi, \mathcal{X}, \mathcal{D}, M)$ Oracle to return the sample S to the learner (as opposed to NSN Oracle in NastyConsistent).

Algorithm MappingErrorTolerantConsistent

Input: certainty parameter $\delta > 0$, the mapping error rate $\beta < \frac{1}{2}$ and required accuracy $\epsilon = 2\eta + \Delta$.

begin

1) Request a labeled dataset $S = \{\langle x, \phi(x) \rangle\}$ of size $m > \frac{c}{\Delta^2}(d + log2/\delta)$ from $kEX^m(\phi, \mathcal{X}, \mathcal{D}, M)$.
2) Output any $h \in \mathcal{F}$ such that $|\{x \in S : h(x) \neq \phi(x)\}| \leq m(\eta + \Delta/4)$ (if no such h exists, output any $h \in \mathcal{F}$).

end

***Theorem** 3:* Let $\mathcal{C}$ be any class of VC-dimension d. Then, there exists a choice of the constant c for which *MappingErrorTolerantConsistent* is a PAC learning algorithm under the mapping error rate β.

Proof: The algorithm *MappingErrorTolerantConsistent* differs *NastyConsistent* in that it uses a $kEX^m(\phi, \mathcal{X}, \mathcal{D}, M)$ instead of a NSN Oracle to get the labeled dataset. Since we have shown that the $kEX^m(\phi, \mathcal{X}, \mathcal{D}, M)$ can be simulated by the NCN oracle which in turn can be simulated by the NSN oracle, the statement of the theorem follows from Theorem 2. ■

The observation that learning in presence of mapping errors is a weaker case of the NSN model results in some open problems. It is known that certain concept-classes are not PAC-learnable in the NCN setting (see Non-trivial Concept class and associated theorem 3 in [6]). This raises the open question as to whether such a non-trivial concept class is PAC learnable in the restricted case of IDCN and hence in the presence of mapping errors

IV. Summary and Discussion

A. Significance

The rapid proliferation of autonomous, distributed data sources in many emerging data-rich domains (e.g., bioinformatics, social informatics, security informatics) coupled with the rise in the use of ontologies to associate semantics with the data has led to a growing interest in the problem of learning predictive models from *semantically disparate* data

sources. Many practical approaches to this problem rely on *mapping* the instance descriptions used by the individual data sources into instance descriptions expressed in a common representation assumed by the learner (As an example [16] lists mappings between 20 different ontologies to the gene ontology). Establishing such mappings is a complex and inevitably error-prone process. Hence there is a need for approaches to learning from such data in the presence of mapping errors. In this paper we have established that the problem of learning from semantically disparate data sources in the presence of mapping errors can be reduced to the problem of learning from a single data source in the presence of nasty classification noise within a PAC-like framework. Based on this reduction, we outlined some of the techniques that can be used to cope with errors in mappings in this setting. We believe these techniques will prove to be very useful in practice as the use of ontologies becomes even more widespread. On a theoretical side, we also presented an algorithm that can be used to learn in presence of mapping errors in a PAC like setting.

B. Related Work

The problem of learning predictive models in the presence of noise in the data has received considerable attention in the literature, specifically in a PAC like setting [21] [22] [5][6]. There is also growing interest in the problem of learning predictive models from distributed data sources [17] [18]. Crammer et al. [19] have examined the problem of learning predictors from a set of *related* data sources. Ben-David et al. [20] have analyzed the sample complexity of learning from semantically disparate data sources. However, none of these works have considered the effect of errors in mappings between the representations used by the individual data sources. Of related interest is the work in ontology mapping field [3] [23]. However, the primary focus in this area is aligning ontologies (through use of mappings), merging related ontologies or detecting logical inconsistencies in mappings [24]. However, a consistent mapping need not be correct in the sense described in this paper and in addition the focus of this paper is to learn in presence of mapping errors.

REFERENCES

[1] M. Lenzerini, "Data integration: a theoretical perspective," in Proceedings of *PODS '02*, pp. 233–246.

[2] R. Hull, "Managing semantic heterogeneity in databases: a theoretical prospective," in Proceedings of *PODS '97*, pp. 51–61.

[3] Y. Kalfoglou and M. Schorlemmer, "Ontology mapping: The state of the art," *Knowl. Eng. Rev*, vol. 18, no. 04391, pp. 1–31, 2005.

[4] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[5] M. J. Kearns and U. V. Vazirani, *An Introduction to Computational Learning Theory*. The MIT Press, August 1994.

[6] N. H. Bshouty, N. Eiron, and E. Kushilevitz, "Pac learning with nasty noise," *Theor. Comput. Sci.*, vol. 2888, p. 2002, 1999.

[7] G. H. John, "Robust decision trees: Removing outliers from databases," in Proceedings of *KDD '95*, pp. 174–179.

[8] Y. Mansour, "Pessimistic decision tree pruning based on tree size," in Proceedings of *ICML '97*, pp. 195–201.

[9] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[10] S. Verbaeten and A. V. Assche, "Ensemble methods for noise elimination in classification problems," in *Multiple Classifier Systems*, 2003, pp. 317–325.

[11] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *ICML*, 2003, pp. 920–927.

[12] D. Gamberger, N. Lavrac, and C. Groselj, "Experiments with noise filtering in a medical domain," in Proceedings of *ICML '99*, pp. 143–151.

[13] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, 2000.

[14] H. Allende-Cid, R. Salas, H. Allende, and R. Ñanculef, "Robust alternating adaboost," in *CIARP*, 2007, pp. 427–436.

[15] "Learnability and the VapnikChervonenkis dimension," Journal of the ACM, vol. 36, no. 4, pp. 929–865, 1989.

[16] "Mappings of external classification systems to gene ontology." [Online]. Available: *http://www.geneontology.org/GO.indices.shtml*

[17] B. Park and H. Kargupta, *The Handbook of Data Mining*. Lawrence Erlbaum Associates, 2003, ch. Distributed Data Mining, pp. 341–358.

[18] D. Caragea, J. Zhang, J. Bao, J. Pathak, and V. Honavar, "Algorithms and software for collaborative discovery from autonomous, semantically heterogeneous information sources (invited paper)," in Proceedings of *ALT 05*, pp. 13–44.

[19] K. Crammer, M. Kearns, and J. Wortman, "Learning from multiple sources," *J. Mach. Learn. Res.*, vol. 9, pp. 1757–1774, 2008.

[20] S. Ben-david, J. Gehrke, and R. Schuller, "A theoretical framework for learning from a pool of disparate data sources," in Proceedings of *KDD 2002*, pp. 443–449.

[21] D. Angluin and P. Laird, "Learning from noisy examples," *Machine Learning*, vol. 2, pp. 343–370, April 1998.

[22] G. Shackelford and D. Volper, "Learning k-dnf with noise in the attributes," in Proceedings of *COLT '88*, pp. 97–103.

[23] J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer-Verlag, 2007.

[24] C. Meilicke, H. Stuckenschmidt, and A. Tamilin, "Repairing ontology mappings," in *AAAI*, 2007, pp. 1408–1413.