Gradual Machine Learning for Entity Resolution (Technical Report)

Boyi Hou, Qun Chen, Yanyan Wang, Youcef Nafa, Zhanhuai Li
School of Computer Science, Northwestern Polytechnical University
Xi’an, Shaanxi
{ntoskrnl@mail.,chenbenben@,wangyanyan@mail.,youcef.nafa@mail.,lizhh@}nwpu.edu.cn

ABSTRACT

Usually considered as a classification problem, entity resolution (ER) can be very challenging on real data due to the prevalence of dirty values. The state-of-the-art solutions for ER were built on a variety of learning models (most notably deep neural networks), which require lots of accurately labeled training data. Unfortunately, high-quality labeled data usually require expensive manual work, and are therefore not readily available in many real scenarios. In this paper, we propose a novel learning paradigm for ER, called *gradual machine learning*, which aims to enable effective machine labeling without the requirement for manual labeling effort. It begins with some easy instances in a task, which can be automatically labeled by the machine with high accuracy, and then gradually labels more challenging instances by iterative factor graph inference. In gradual machine learning, the hard instances in a task are gradually labeled in small stages based on the estimated evidential certainty provided by the labeled easier instances. Our extensive experiments on real data have shown that the performance of the proposed approach is considerably better than its unsupervised alternatives, and highly competitive compared to the state-of-the-art supervised techniques. Using ER as a test case, we demonstrate that gradual machine learning is a promising paradigm potentially applicable to other challenging classification tasks requiring extensive labeling effort.

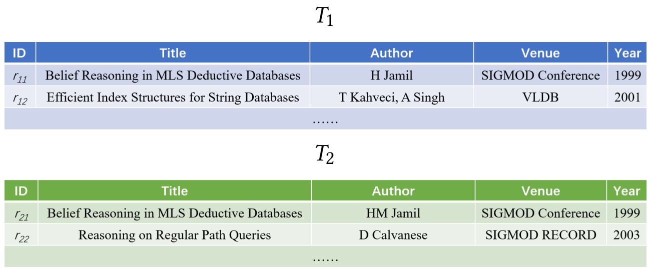
KEYWORDS

Gradual Machine Learning; Entity Resolution; Unsupervised Learning; Factor Graph Inference

1 INTRODUCTION

The task of entity resolution (ER) aims at finding the records that refer to the same real-world entity [15]. Consider the running example shown in Figure 1. ER needs to match the paper records between two tables, T_1 and T_2. The pair of $< r_{1i}, r_{2j} >$, in which r_{1i} and r_{2j} denote a record in T_1 and T_2 respectively, is called an *equivalent* pair if and only if r_{1i} and r_{2j} refer to the same paper; otherwise, it is called an *inequivalent* pair. In the example, r_{11} and r_{21} are *equivalent* while r_{11} and r_{22} are *inequivalent*. The state-of-the-art solutions for ER were built on a variety of learning models (e.g. deep neural network (DNN) [34]), which require lots of accurately labeled training data. Unfortunately, high-quality labeled data usually require expensive manual work, and therefore, may not be readily available in many real scenarios.

It can be observed that the dependence of the existing supervised learning models on high-quality labeled data is not limited to the task of ER. The dependence is actually crucial for their huge success in various domains (e.g. image and speech recognition [49]).



T_1

ID	Title	Author	Venue	Year
r_{11}	Belief Reasoning in MLS Deductive Databases	H Jamil	SIGMOD Conference	1999
r_{12}	Efficient Index Structures for String Databases	T Kahveci, A Singh	VLDB	2001
……				

T_2

ID	Title	Author	Venue	Year
r_{21}	Belief Reasoning in MLS Deductive Databases	HM Jamil	SIGMOD Conference	1999
r_{22}	Reasoning on Regular Path Queries	D Calvanese	SIGMOD RECORD	2003
……				

Figure 1: An ER Example

However, it has been well recognized that in some real scenarios, where high-quality labeled data is scarce, their efficacy can be severely compromised. To address the limitation resulting from such dependence, we propose a novel learning paradigm, called *gradual machine learning*, in which *gradual* means proceeding in small stages. Gradual machine learning aims to enable effective machine labeling without the requirement for manual labeling effort. Inspired by the gradual nature of human learning, which is adept at solving the problems with increasing hardness, it begins with some easy instances in a task, which can be automatically labeled by the machine with high accuracy, and then gradually reasons about the labels of the more challenging instances based on the observations provided by the labeled easier instances.

We note that there already exist many learning paradigms for a variety of classification tasks, including transfer learning [35], lifelong learning [13], curriculum learning [5], self-paced learning [27] and self-training learning [30] to name a few. Transfer learning focused on using the labeled training data in a domain to help learning in another target domain. Lifelong learning studied how to leverage the knowledge mined from past tasks for the current task. Curriculum learning investigated how to organize a curriculum (the presenting order of training examples) for improved model training. Self-training learning aimed to improve the performance of a supervised learning algorithm by incorporating unlabeled data into the training data set. More recently, Snorkel [37] aimed to enable automatic and massive machine labeling by specifying a wide variety of labeling functions. The results of machine labeling were supposed to be fed to DNN for model training. However, the following two properties of gradual machine learning make it fundamentally different from the existing learning paradigms:

- Distribution misalignment between easy and hard instances in a task. Gradual machine learning processes the instances in the increasing order of hardness. Its scenario does not satisfy the i.i.d (independent and identically distributed) assumption underlying most existing machine learning models: the labeled easy instances are not representative of the unlabeled harder instances. The distribution misalignment between the labeled and unlabeled instances renders most existing learning models unfit for gradual machine learning.

- Gradual learning by small stages in a task. Gradual machine learning proceeds in small stages. At each stage, it typically labels only one instance based on the evidential certainty provided by the labeled easier instances. The process of iterative labeling can be performed in an unsupervised manner without requiring any human intervention.

We summarize the major contributions of this paper as follows:

(1) We propose a novel learning paradigm of Gradual Machine Learning (GML), which can effectively eliminate the requirement for manual labeling effort for the challenging classification tasks;
(2) We present a technical solution based on the proposed paradigm for entity resolution. We present a package of techniques, including easy instance labeling, feature extraction and influence modeling, and gradual inference, to enable effective gradual machine learning for ER.
(3) Our extensive experiments on real data have validated the efficacy of the proposed approach. Our empirical study has shown that the performance of the proposed approach is considerably better than the unsupervised alternatives, and highly competitive compared to the state-of-the-art supervised techniques. It also scales well with workload size.

Note that a prototype of the proposed GML solution for ER has been presented in the demo paper of [22]. Besides providing with more technical details on GML for ER, this technical paper makes the following new contributions:

(1) We propose a scalable approach for gradual inference. The general approach consists of three steps, measurement of evidential support, approximate estimation of inference probability, and construction of inference subgraph.
(2) We present the algorithms for the three steps of the scalable approach to enable efficient gradual inference.
(3) We evaluate the performance sensitivity of the proposed solution w.r.t various algorithmic parameters and its scalability. Our experimental results have shown that the proposed solution performs robustly w.r.t the parameters and it scales well with workload size.

It is also noteworthy that we have recently applied the GML paradigm on the task of aspect-level sentiment analysis [46]. Similar to the task of ER, the performance of GML has been shown to be highly competitive compared to the state-of-the-art DNN techniques.

The rest of this paper is organized as follows: Section 2 reviews related work. Section 3 defines the task of ER. Section 4 introduces the GML paradigm. Section 5 proposes the technical solution for ER. Section 6 presents the solution of scalable gradual inference for ER. Section 7 presents our empirical evaluation results. Finally, Section 8 concludes this paper.

2 RELATED WORK

In this section, we review related work from the orthogonal perspectives of machine learning and entity resolution.

2.1 Machine Learning Paradigms

Note that many machine learning paradigms have been proposed for a wide variety of classification tasks. Here, our intention is not to exhaustively review all the work. We instead review those closely related to our work and emphasize their difference from gradual machine learning.

Traditional supervised machine learning algorithms make predictions on the future data using statistical models that are trained on previously collected labeled training data [14]. In many real scenarios, the labeled data may be too few to build a good classifier. Semi-supervised learning [7, 23] addresses this problem by making use of a large amount of unlabeled data and a small amount of labeled data. Similarly, as an autonomous supervised learning approach, self-supervised learning [31] usually extracts and uses the naturally available relevant context and embedded meta data as supervisory signals. Active learning [3, 4] is another special case of supervised learning in which a learning algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at new data points. The main advantage of active learning over traditional supervised learning is that it usually requires less labeled data for model training. Online learning [25] and incremental learning [39] have also been proposed for the scenarios where training data only becomes available gradually over time or its size is out of system memory limit. Nevertheless, the efficacy of the aforementioned learning paradigms depends on the i.i.d assumption. Therefore, they can not be applied to the scenario of gradual machine learning.

Curriculum learning (CL) [5] and self-paced learning (SPL) [27] are to some extent similar to gradual machine learning in that they were also inspired by the learning principle underlying the cognitive process in humans, which generally starts with learning easier aspects of a task, and then gradually takes more complex examples into consideration. Both of them essentially investigated how to feed model training with a sequence of samples ranked by learning difficulty for improved performance. The difference is that curriculum learning mainly focused on how to pre-organize a curriculum (the presenting order of training examples), while self-paced learning proposed to insert a regularizer into the training objective function to automatically optimizing the presenting order in the training process. However, the models trained by curriculum learning or self-paced learning are supposed to be applied on a target workload satisfying the i.i.d assumption. Therefore, as traditional supervised learning, their efficacy still depends on good-quality training examples. More recently, some researchers proposed the approach of self-paced deep clustering for image classification [10, 21]. Iteratively alternating between deep representation learning and clustering, it essentially used self-paced learning to improve deep representation for better clustering performance. At each iteration, a deep representation model is first trained in a self-paced manner based on the clustering results, and the resulting model is then applied to generate the deep representations for the instances in a target workload. It can be observed that similar to traditional self-paced learning, the efficacy of self-paced representation learning still depends on the i.i.d assumption. On the other hand, self-paced deep clustering used the classical clustering algorithms (eg. k-means) to label the instances based on their learned representations in a batch manner. Therefore, self-paced deep clustering is at its core a clustering approach. In contrast, gradual machine learning gradually reasons about the labels of the hard instances by factor graph inference without the assumption of i.i.d. It does not need any clustering algorithm.

In contrast, transfer learning [35], allows the distributions of the data used in training and testing to be different. The other learning techniques closely related to transfer learning include lifelong learning [13] and multi-task learning [8]. Different from transfer learning, lifelong learning usually assumes that the current task has good training data, and aims to further improve the learning using both the target domain training data and the knowledge gained in past learning. Multi-task learning instead tries to learn multiple tasks simultaneously even when they are different. However, these learning paradigms can not be applied to the scenario of gradual machine learning either. Firstly, focusing on unsupervised learning within a task, gradual machine learning does not enjoy the access to good labeled training data or a well-trained classifier to kick-start learning. Secondly, the existing techniques transfer instances or knowledge between tasks in a batch manner; they do not support gradual learning by small stages on the instances with increasing hardness within a task.

2.2 Work on Entity Resolution

Research effort on unsupervised entity resolution were mainly dedicated to devising various distance functions to measure pair-wise similarity [32]. However, it has been empirically shown [6] that the efficacy of these unsupervised techniques is limited. Alternatively, ER can be automatically performed based on rules [18, 29, 42], probabilistic theory [19, 43] and machine learning [14, 17, 26, 38]. Compared with the unsupervised alternatives, they can effectively improve the quality of entity resolution to some extent. However, good performance of these supervised techniques depends on the presence of effective rules or a large quantity of accurately labeled training data, which may not be readily available in real applications. To reduce the cost of data labeling, many active learning techniques [33, 38] have been proposed for the task of ER. Active learning has also been leveraged to ensure a pre-specified precision requirement for ER [3, 4].

The progressive paradigm for ER [2, 47] has also been proposed for the scenario in which ER should be processed efficiently but does not necessarily require to generate high-quality results. Taking a pay-as-you-go approach, it studied how to maximize result quality given a pre-specified resolution budget. However, the target scenario of progressive ER is different from that of gradual machine learning, whose major challenge is to label the instances with increasing hardness without resolution budget.

It has been well recognized that pure machine algorithms may not be able to produce satisfactory results in practical scenarios [28]. Therefore, many researchers [9, 16, 20, 33, 44, 45, 48] have studied how to crowdsource an ER workload. While these researchers addressed the challenges specific to crowdsourcing, we instead investigate a different problem in this paper: how to enable unsupervised gradual machine learning.

3 TASK STATEMENT

ER reasons about the equivalence between two records. Two records are deemed to be equivalent if and only if they correspond to the same real-world entity. Given an ER workload consisting of record pairs, a solution labels each pair in the workload as *matching* or *unmatching*.

Table 1: Frequently Used Notations.

Notation	Description
D	an ER workload consisting of record pairs
D_i	a subset of D
S	a labeling solution for D
d, d_i	a record pair in D
$P(d_i)$	the estimated equivalence probability of d_i
f, f_i	a feature of record pair
F, F_i	a feature set
D_f	the set of record pairs having the feature f

For the sake of presentation simplicity, we summarize the frequently used notations in Table. 1. As usual, we measure the quality of a labeling solution by the unified metric of F-1, which can be represented by

$$f_1(D,S) = \frac{2}{\frac{1}{precision(D,S)} + \frac{1}{recall(D,S)}}. \quad (1)$$

in which $precision(D,S)$ and $recall(D,S)$ denote the achieved precision and recall of S on D respectively.

Finally, the task of entity resolution is defined as follows:

Definition 3.1. **[Entity Resolution].** Given a workload consisting of record pairs, $D = \{d_1, d_2, \cdots, d_n\}$, the task of entity resolution is to give a labeling solution S for D such that $f_1(D,S)$ is maximized.

4 LEARNING PARADIGM

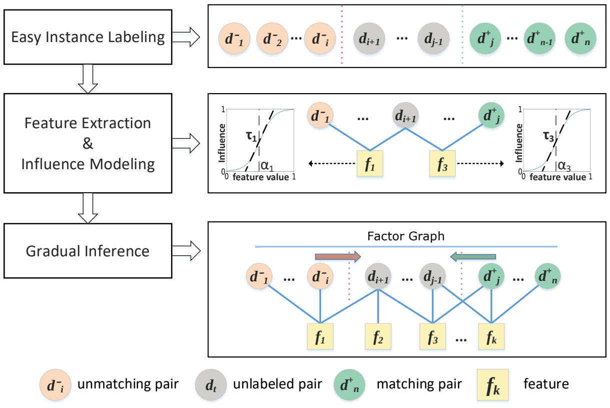

Easy Instance Labeling
d^-_1 d^-_2 ··· d^-_i d_{i+1} ··· d_{j-1} d^+_j ··· d^+_{n-1} d^+_n
Feature Extraction & Influence Modeling
Influence
1
0
τ_1
α_1
0 feature value 1
d^-_1 ... d_{i+1} ... d^+_j
f_1
f_3
τ_3
α_3
Gradual Inference
Factor Graph
d^-_1 ... d^-_i d_{i+1} ... d_{j-1} d^+_j ... d^+_n
f_1 f_2 f_3 ... f_k
d^-_i unmatching pair
d_t unlabeled pair
d^+_n matching pair
f_k feature

Figure 2: Paradigm Overview.

The process of gradual machine learning, as shown in Figure 2, consists of the following three essential steps:

- **Easy Instance Labeling.** Given a classification task, it is usually very challenging to accurately label all the instances in the task without good-coverage training examples. However, the work can become much easier if we only need to automatically label some easy instances in the task. In the case of ER, while the pairs with the medium similarities are usually challenging for machine labeling, highly similar (resp. dissimilar) pairs have fairly high probabilities to be equivalent (resp. inequivalent). They can therefore be chosen as easy instances. In real scenarios, easy instance labeling can be performed based on the simple user-specified rules or the existing unsupervised learning techniques. Gradual machine learning begins with the observations provided by the labels of easy instances. Therefore, the high accuracy of automatic machine labeling on easy instances is critical for its ultimate performance on a given task.
- **Feature Extraction and Influence Modeling.** Features serve as the medium to convey the knowledge obtained from the labeled easy instances to the unlabeled harder ones. This step extracts the common features shared by the labeled and unlabeled instances. To facilitate effective knowledge conveyance, it is desirable that a wide variety of features are extracted to capture as much information as possible. For each extracted feature, this step also needs to model its influence over the labels of its relevant instances.
- **Gradual Inference.** This step gradually labels the instances with increasing hardness in a task. Since the scenario of gradual learning does not satisfy the i.i.d assumption, we propose to fulfill gradual learning from the perspective of evidential certainty. As shown in Figure 2, we construct a factor graph, which consisting of the labeled and unlabeled instances and their common features. Gradual learning is conducted over the factor graph by iterative factor graph inference. At each iteration, it chooses an unlabeled instance for labeling. The iteration is repeatedly invoked until all the instances in a task are labeled. Note that in gradual inference, a newly labeled instance at the current iteration would serve as an evidence observation in the following iterations.

Since gradual machine learning is characterized by gradual inference, we formulate the process of gradual inference. Formally, we denote the model of factor graph corresponding to a classification workload by G. Suppose that G consists of a set of evidence variables $\mathbf{\Lambda}$, whose labels are known, a set of inference variables $\mathbf{X}$, whose labels are unknown, and a group of factor functions of variables to indicate the probabilistic relations among the variables, denoted by $\mathbf{F}_\theta(D_i) : D_i \to P_\theta(D_i)$, in which D_i denotes a set of variables and $D_i \in \text{PowerSet}(\mathbf{\Lambda} \cup \mathbf{X})$.

Gradual inference iteratively labels an inference variable $x_i \in \mathbf{X}$ by factor graph inference until all the inference variables in G are labeled. In each iteration, GML generally chooses to label the inference variable with the highest degree of evidential certainty. Suppose that the total number of label types, denoted by $\{L_1, L_2, \ldots, L_l\}$, is l. Given an instance d, GML measures its evidential certainty by the inverse of entropy [41] as follows

$$E(d) = \frac{1}{H(d)} = \frac{1}{-\sum\limits_{1 \le i \le l} P_i(d) \cdot \log_2 P_i(d)}, \tag{2}$$

in which $E(d)$ and $H(d)$ denote the evidential certainty and entropy of d respectively, and $P_i(d)$ denotes the inferred probability of d having the label of L_i.

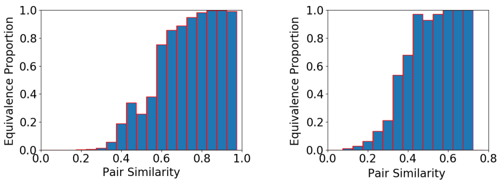
5 SOLUTION FOR ER

5.1 Easy Instance Labeling

Given an ER workload, the solution identifies the easy instances by the simple rules specified on record similarity. The set of easy instances labeled as *matching* is generated by setting a high lowerbound on record similarity. Similarly, the set of easy instances labeled as *unmatching* is generated by setting a low upperbound on record similarity. To explain the effectiveness of the rule-based approach, we introduce the monotonicity assumption of precision, which was first defined in [3], as follows:

Assumption 1 (Monotonicity of Precision). *A value interval I_i is dominated by another interval I_j, denoted by $I_i \preceq I_j$, if every value in I_i is less than every value in I_j. We say that precision is monotonic with respect to a pair metric if for any two value intervals $I_i \preceq I_j$ in [0,1], we have $P(I_i) \le P(I_j)$, in which $P(I_i)$ denotes the equivalence precision of the set of instance pairs whose metric values are located in I_i.*

According to the monotonicity assumption, we can *statistically* state that a pair with a high (resp. low) similarity has a correspondingly high probability of being an equivalent (resp. inequivalent) pair. These record pairs can be deemed to be easy in that they can be automatically labeled by the machine with high accuracy. In comparison, the instance pairs having the medium similarities are more challenging because labeling them either way by the machine would introduce considerable errors.



(a) DBLP-Scholar **(b) abt-buy**

Figure 3: Empirical Validation of the Monotonicity Assumption.

We have empirically validated the monotonicity assumption on the real datasets of DBLP-Scholor[1] and Abt-Buy[2]. The precision levels of different similarity intervals are shown in Figure 3. It can be observed that statistically speaking, precision increases with similarity value with notably rare exceptions. It is noteworthy that given a machine metric for a classification task, the monotonicity assumption of precision actually underlies its effectiveness as a classification metric. Therefore, the easy instances in an ER task can be similarly identified by other classification metrics.

In the scenario of ER, we suppose to identify 30%-50% of the instances in a workload as easy, in order to provide an accurate

[1] available at https://dbs.uni-leipzig.de/file/DBLP-Scholar.zip

[2] available at https://dbs.uni-leipzig.de/file/Abt-Buy.zip

and also good-coverage initial labeling knowledge. From the monotonicity assumption we can know that, the labels of easy instances are always more accurate than hard ones. Considering the common case that the similarity equals the probability and are uniformly distributed on the $[0, 1]$ interval, therefore, when labeling the easy instances whose similarities are at the two extreme sides, the expectation of their mislabeled ratio ε with regard to the easy instances ratio e is $\varepsilon(e) = \frac{2}{e} \cdot \int_{0}^{\frac{e}{2}} x dx = \frac{e}{4}$. Within a tolerable initial mislabeled ratio, we will expect e as larger as possible to cover more labeling knowledge. It can be seen that when $e = 0.3$, ε is still a small value less that 0.1, and when $e = 0.5$, ε is still a small value close to 0.1.

5.2 Feature Extraction and Influence Modeling

The guiding principle of feature extraction is to extract a wide variety of discriminating features that can capture as much information as possible from the record pairs. For ER, we extract the following two types of features from record pairs:

(1) Attribute value similarity. This type of feature measures a pair's value similarity at each record attribute. Different attributes may require different similarity metrics.
(2) Token feature. We denote a token by o_i, the feature that o_i occurs in both records by $Same(o_i)$ and the feature that o_i occurs in one and only one record by $Diff(o_i)$. Note that the feature of $Same(o_i)$ serves as evidence for equivalence, while $Diff(o_i)$ indicates the opposite. Unlike attribute value similarity, which treats attribute values as a whole, token feature considers the influence of each individual token on equivalence probability. For the workloads with miscellaneous tokens, not every token is highly discriminating (or indicative of entity identity); therefore, we filter the tokens by the metric of IDF (inverse document frequency).

It is worthy to point out that attribute similarity metrics have been extensively studied in the literature [12]. In GML, given an attribute type, we simply select the metrics which have been empirically shown to be effective in indicating equivalence status. For instance, on DBLP-Scholar, the appropriate metric for the *venue* attribute is the edit distance, while the appropriate metric for the *title* attribute is instead a hybrid metric combining Jaccard similarity and edit distance. For the attribute of *title*, we also use the metric of longest common substring because it has been widely used to capture the similarity between two order-sensitive long token strings. It is noteworthy that given an attribute type, its similarity metrics can be applied on any pair of particular values. As a result, the features of similarity metrics are usually shared by all the pair instances provided that their corresponding attribute values are not null.

For dataset with miscellaneous tokens such as Abt-Buy, since not every token is highly discriminating (or indicative of entity identity), we filter the tokens in a workload by the metric of IDF (inverse document frequency). Specifically, only the tokens, whose IDF value are within a pre-specified range (eg. $[\ln(\frac{1}{3}N), \ln(\frac{1}{2}N)]$, N denotes the total records in the dataset), are extracted as features. It can be observed that if a token occurs too frequently, it usually has a limited capability to indicate entity identity; on the other hand, if it is a rare token, its few occurrences may render it almost useless for gradual inference. By this filtering mechanism, we effectively ensure that any token feature is not only to some extent indicative of entity identity, but helpful to gradual inference.

The aforementioned two types of features can provide a good coverage of the discriminating information contained in record pairs. We observe that both types of features can be supposed to satisfy the monotonicity assumption of precision. Therefore, as shown in Figure 4, for each feature, we model its influence over pair labels by a monotonous sigmoid function with two parameters, α and τ, which denote the x-value of the function's midpoint and the steepness of the curve respectively. The x-value of the sigmoid function represents the feature values of pairs, and the y-value represents their equivalent probabilities as indicated by the feature. Formally, given a feature f and a pair d, the influence of f w.r.t d is represented by

$$P_f(d) = \frac{1}{1 + e^{-\tau_f(x_f(d) - \alpha_f)}}, \tag{3}$$

in which $x_f(d)$ represents d's feature value w.r.t f. According to Eq. 3, provided with the values of α_f and τ_f, the influence model statistically dictates that any feature value of $x_f(d)$ corresponds to an equivalence probability. Typically, the value of $P_f(d)$ increases with the feature value of d, or $x_f(d)$. As illustrated by the examples shown in Figure 4, different combinations of α_f and τ_f can result in different influence model shapes. Note that since the second type of features has the constant value of 1, we first align them with record similarity and then model their influence by sigmoid functions.

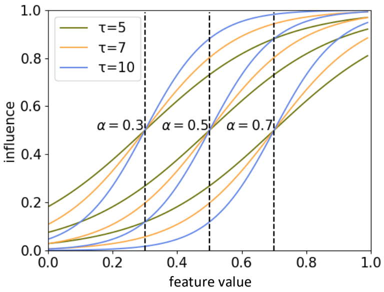


Figure 4: the Examples of Sigmoid Function.

It is noteworthy that given a sigmoid model, gradual machine learning essentially reasons about the labels of the middle points, which correspond to the hard instances, provided with the labels of the more extreme points at both sides, which correspond to the easy instances. If it were not for the monotonicity assumption, estimating the labels of the middle points by regression would be too erroneous because the more extreme observations at both sides are not their valid representatives. Our solution overcomes this hurdle by assuming monotonicity of precision and proceeding in small stages, in each of which the regression results of only a few instances close to the labeled easy instances are considered for equivalence reasoning. Fortunately, monotonicity of precision is a universal assumption underlying the effectiveness of the existing machine metrics for classification tasks. Therefore, our proposed solution for modeling feature influence can be potentially generalized for other classification tasks.

5.3 Gradual Inference

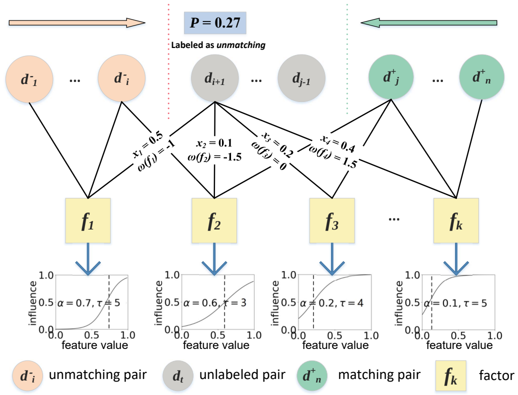


Figure 5: An Example of Factor Graph.

To enable gradual machine learning, we construct a factor graph, G, which consists of the labeled easy instances, the unlabeled hard instances and their common features. In G, the labeled easy instances are represented by the *evidence variables*, the unlabeled hard instances by the *inference variables*, and the features by the *factors*. The value of each variable represents its corresponding pair's equivalence probability. An evidence variable has the constant value of 0 or 1, which indicates the status of *unmatching* and *matching* respectively. It participates in gradual inference, but its value remains unchanged during the inference process.

An example of factor graph is shown in Figure 5. Each variable has multiple factors, each of which corresponds to a feature. Since a feature can be shared among multiple pairs, for presentation simplicity, we represent a feature by a single factor and connect it to multiple variables. Note that given a feature f and a pair d, the influence of f w.r.t d is represented by the sigmoid function of

$$P_f(d) = \frac{1}{1 + e^{-\tau_f(x_f(d) - \alpha_f)}}, \tag{4}$$

in which $x_f(d)$ represents f's value w.r.t d, which is known beforehand, and τ_f and α_f represent the parameters of a sigmoid function, which need to be learned. Accordingly, in the factor graph, we represent the factor weigh of f w.r.t d by

$$\omega_f(d) = \theta_f(d) \cdot \log(\frac{P_f(d)}{1 - P_f(d)}) = \theta_f(d) \cdot \tau_f(x_f(d) - \alpha_f), \tag{5}$$

in which $\log(\cdot)$ codes the estimated influence of f on d by sigmoid regression, and $\theta_f(d)$ represents the confidence on influence estimation. In practical implementation, we can estimate $\theta_f(d)$ based on the theory of regression error bound [11]. More details on the estimation of $\theta_f(d)$ will be discussed in Subsection 6.1.

Denoting the feature set of a pair d by F_d, a factor graph infers the equivalence probability of d, $P(d)$, by:

$$P(d) = \frac{\prod\limits_{f \in F_d} e^{\omega_f(d)}}{1 + \prod\limits_{f \in F_d} e^{\omega_f(d)}}. \tag{6}$$

The process of gradual inference essentially learns the parameter values (α and τ) of all the features such that the inferred results maximally match the evidence observations on the labeled instances. Formally, the objective function can be represented by

$$(\hat{\alpha}, \hat{\tau}) = \arg\min_{\alpha,\tau} - \log \sum_{V_I} P_{\alpha,\tau}(\Lambda, V_I), \tag{7}$$

in which Λ denotes the observed labels of evidence variables, V_I denotes the inference variables in G, and $P_{\alpha,\tau}(\Lambda, V_I)$ denotes the joint probability of the variables in G. Since the variables in G are conditionally independent, $P_{\alpha,\tau}(\Lambda, V_I)$ can be represented by:

$$P_{\alpha,\tau}(\Lambda, V_I) = \prod_{d \in \Lambda \cup V_I} P_{\alpha,\tau}(d). \tag{8}$$

Accordingly, the objective function can be simplified into

$$(\hat{\alpha}, \hat{\tau}) = \arg\min_{\alpha,\tau} - \sum_{d \in \Lambda} \log P_{\alpha,\tau}(d). \tag{9}$$

Considering the unbalanced populations of two classes, we weight the observations of two classes to perform the weighted maximum likelihood estimation as in [1, 24]. The approach essentially weights positive and negative observations by the inverses of their total occurrences. Specifically, given a factor graph consisting of n_- unmatching and n_+ matching observations, we set the weights of the unmatching and matching observations as 1 and $\frac{n_-}{n_+}$ respectively. Accordingly, the objective function can be finally represented by

$$(\hat{\alpha}, \hat{\tau}) = \arg\min_{\alpha,\tau} - \sum_{d \in \Lambda} t_d \cdot \log P_{\alpha,\tau}(d), \tag{10}$$

in which $t_d = 1$ if d is labeled as unmatching, and $t_d = \frac{n_-}{n_+}$ if d is labeled as matching.

Given a factor graph, G, at each stage, gradual inference first reasons about the parameter values of the features and the equivalence probabilities of the unlabeled pairs by maximum likelihood, and then labels the unlabeled pair with the highest degree of evidential certainty. Note that GML defines evidential certainty as the inverse of entropy. Formally, in the case of ER, evidential certainty is measured by

$$E(d) = \frac{1}{-(P(d) \cdot \log_2 P(d) + (1 - P(d)) \cdot \log_2(1 - P(d)))}, \tag{11}$$

in which $E(d)$ denotes the evidential certainty of d.

6 SCALABLE GRADUAL INFERENCE

It can be observed that repeated inference by maximum likelihood estimation over a large-sized factor graph of the whole variables is usually very time-consuming [50]. As a result, there is a need for efficient gradual inference that can scale well with large workloads. In this section, we present a scalable approach that can effectively fulfill gradual learning without repeatedly inferring over the entire factor graph.

Algorithm 1: Scalable Gradual Inference

```
while there exists any unlabeled variable in G do
    $V' \leftarrow$ all the unlabeled variables in $G$;
    for $v \in V'$ do
        Measure the evidential support of $v$ in $G$;
    end
    Select top-m unlabeled variables with the most
      evidential support (denoted by $V_m$) ;
    for $v \in V_m$ do
        Estimate the probability of $v$ in $G$ by approximation;
    end
    Select top-$k$ certain variables in terms of entropy in $V_m$
      based on the approximate probabilities (denoted by $V_k$)
      ;
    for $v \in V_k$ do
        Compute the probability of $v$ in $G$ by the factor
          graph inference over a subgraph of $G$;
    end
    Label the variable with the minimal entropy in $V_k$;
end
```

The scalable solution is crafted based on the following observations:

- Many unlabeled inference variables in the factor graph may be only weakly linked through the factors to the evidence variables. Due to lack of evidential support, their inferred probabilities would be quite ambiguous, i.e. close to 0.5. As a result, at each stage, only the inference variables that have received considerable support from the evidence variables need to be considered for labeling;
- With regard to the probability inference of a single variable v in a large factor graph, it can be effectively approximated by considering the potentially much smaller subgraph consisting of v and its neighboring variables. The inference over the subgraph can usually be much more efficient than over the original entire graph.

The process of scalable gradual inference is sketched in Algorithm 1. It first selects the top-m unlabeled variables with the most evidential support in G as the candidates for probability inference. To reduce the invocation of maximum likelihood estimation, it then approximates probability inference by an efficient algorithm on the m candidates. Finally, it infers via maximum likelihood the probabilities of only the top-k most promising unlabeled variables among the m candidates. For each variable in the final set of k candidates, its probability is not inferred over the entire graph of G, but over a potentially much smaller subgraph. In the rest of this section, we will present the technique for each of the three steps.

6.1 Measurement of Evidential Support

Since the influence of a feature over the pairs is modeled by a sigmoid function, we consider the evidential support that an unlabeled variable receives from a feature as the confidence on the regression result provided by its corresponding function, denoted by $\theta_f(d)$. Given an unlabeled variable, d, we first estimate its evidential support provided by each of its factors based on the theory of regression error bound [11], and then aggregate them to estimate its overall evidential support based on the Dempster-Shafer theory [40].

Formally, for the influence estimation of a single feature f on the variables, the process of parameter optimization corresponds to a linear regression between the natural logarithmic coded influence in Eq. 5, hereinafter denoted by $l_f(d)$, and the feature value $x_f(d)$, as follows

$$l_f(d) = \tau_f \cdot x_f(d) - \tau_f \cdot \alpha_f + \varepsilon, \tag{12}$$

in which ε denotes the regression residual. The parameters α_f and τ_f are optimized by minimizing the regression residual as follows:

$$(\hat{\alpha}_f, \hat{\tau}_f) = \arg\min_{\alpha_f,\tau_f} \sum_{d\in\Lambda_f} t_d \cdot (l_f(d) - (\tau_f \cdot x_f(d) - \tau_f \cdot \alpha_f))^2, \tag{13}$$

in which Λ_f denotes the set of labeled pairs having the feature f. As in Eq. 10, t_d denotes the weights of matching and unmatching observations.

According to the theory of linear regression error bound, given a pair d, its prediction error bound $\delta(l_f(d))$ and the confidence level $\theta_f(d)$ satisfy the following formula

$$\delta(l_f(d)) = \\ t_{(1-\theta_f(d))/2}(|\Lambda_f| - 2) \cdot \hat{\sigma}^2 \cdot \sqrt{1 + \frac{1}{n} + \frac{(x_f(d) - \bar{x}_f)^2}{\sum\limits_{d_i\in\Lambda_f} (x_f(d_i) - \bar{x}_f)^2}}, \tag{14}$$

in which $t_{(1-\theta_f(d))/2}(|\Lambda_f| - 2)$ represents the Student's t-value with $|\Lambda_f| - 2$ degree of freedom at $(1 - \theta_f(d))/2$ quantile, and

$$\hat{\sigma}^2 = \frac{1}{|\Lambda_f| - 2} \sum_{d_i\in\Lambda_f} (l_f(d_i) - (\hat{\tau}_f \cdot x_f(d_i) - \hat{\tau}_f \cdot \hat{\alpha}_f))^2, \tag{15}$$

and

$$\bar{x}_f = \frac{1}{|\Lambda_f|} \sum_{d_i\in\Lambda_f} x_f(d_i). \tag{16}$$

Given an error bound of $\delta(l_f(d))$, we measure the evidential support of an unlabeled variable d provided by f by estimating its corresponding regression confidence level $\theta_f(d)$ according to Eq. 14. Then, we use the Dempster-Shafer (D-S) theory [40] to arrive at a degree of belief that takes into account all the available evidences. Given an unlabeled variable v, the evidential support provided by a feature f can be considered to be the extent that f supports the inference on the value of v: a value of 1 means complete support while a value of 0 corresponds to the lack of any support. Suppose that v has l features, $\{f_1,\cdots,f_l\}$, and the evidential support v receiving from f_i is denoted by θ_i. We first normalize the values of θ_i by $\frac{1+\theta_i}{2}$ so that they fall into the range of $[0.5, 1]$. Then, according to the Dempster's rule, the evidential support of v provided by its features can be represented by

$$\theta_v = \frac{\prod\limits_{1\leq i\leq l} \theta_i}{\prod\limits_{1\leq i\leq l} \theta_i + \prod\limits_{1\leq i\leq l} (1 - \theta_i)}. \tag{17}$$

On time complexity, the total cost of evidential support measurement can be represented by $O(n^2 \cdot n_f)$, in which n denotes the total number of instances in a task and n_f denotes the total number of extracted features. Formally, let Z be the universal set representing all possible states of a system under consideration. By a function of Basic Belief Assignment (BBA), the D-S theory assigns a belief mass to each element of the power set. The mass of an element E_i, $m(E_i)$, expresses the proportion of all relevant and available evidence that supports the claim that the actual state belongs to E_i but to no particular subset of E_i. The masses of elements satisfy

$$\sum_{E_i \in 2^Z} m(E_i) = 1,$$

and

$$m(\emptyset) = 0.$$

Note that if only singleton propositions are assigned belief masses, a BBA function reduces to a classical probability function.

The kernel of *D-S* theory is Dempster's rule, which is rooted in probability theory and constitutes a conjunctive probabilistic inference process. It adopts the orthogonal sum operation to combine evidence, which is rooted in calculating the joint probability of independent events. With two pieces of independent evidence represented by two *BBAs* m_1 and m_2 respectively, the joint mass of a proposition E is calculated in the following manner:

$$m_{1,2}(E) = \frac{\sum_{E_i \cap E_j = E \neq \emptyset} m_1(E_i) \cdot m_2(E_j)}{1 - \sum_{E_i \cap E_j = \emptyset} m_1(E_i) \cdot m_2(E_j)},$$

and

$$m_{1,2}(\emptyset) = 0.$$

in which $\sum_{E_i \cap E_j = \emptyset} m_1(E_i) \cdot m_2(E_j)$ measures the amount of conflict between the two mass sets.

6.2 Approximate Estimation of Inferred Probability

To reduce the prohibitive cost of factor graph inference, there is a need to efficiently approximate the inferred probabilities of these top-m variables such that only a small portion (top-k) of them needs to be inferred using factor graph inference.

As previously mentioned, the feature's natural logarithmic influence w.r.t a pair can be estimated by the linear regression value based on Eq. 12. Therefore, we approximate the factor weight of f w.r.t d, $\hat{\omega}_f(d)$, by

$$\hat{\omega}_f(d) = \theta_f(d) \cdot \hat{\tau}_f(x_f(d) - \hat{\alpha}_f), \tag{18}$$

in which $\theta_f(d)$ represents f's normalized confidence level on the regression result w.r.t d and $\hat{\tau}_f$, and $\hat{\alpha}_f$ are the regression parameter values estimated by Eq. 13. Accordingly, a pair's equivalence probability can be approximated by leveraging the approximate factor weights of all its features as follows

$$\hat{P}(d) = \frac{\prod\limits_{f \in F_d} e^{\hat{\omega}_f(d)}}{1 + \prod\limits_{f \in F_d} e^{\hat{\omega}_f(d)}}, \tag{19}$$

in which F_d denotes the feature set of d.

Accordingly, the entropy of d can be approximated by

$$\hat{H}(d) = -(\hat{P}(d) \cdot \log_2 \hat{P}(d) + (1 - \hat{P}(d)) \cdot \log_2 (1 - \hat{P}(d))). \tag{20}$$

In the initial stages, factor graph inference can even be saved if the entropy of the top variable in the k candidates is considerably smaller than that of any other variable.

In practical implementation, due to high efficiency of evidential support measurement and inference probability approximation, the number of candidate inference variables selected for approximate probability estimation (m) can be set to a large value provided that the selected variables receive considerable support. In the case of ER, we set the threshold of evidential support at 0.9. It means that, we have the combined confidence level of at least 0.9 that a candidate variable can be inferred within the specified error bound based on linear regression by its features. By this threshold, the value of m should be set to be in the order of thousands on our test workloads. On the other hand, the proposed approximation technique can usually provide with an accurate ranking on inference probability. Therefore, considering inefficiency of factor graph inference, we suggest to set the number of candidate inference variables chosen for factor graph inference (k) to a much smaller value, or in the order of tens. Our empirical evaluation in Section 7 has showed that to a large extent, the performance of scalable gradual inference is not sensitive to the parameter settings of m and k.

On time complexity, the total cost of approximate probability estimation can be represented by $O(n \cdot n_f \cdot m)$, in which n_f denotes the total number of extracted features.

6.3 Construction of Inference Subgraph

Factor inference over a large graph is usually very time-consuming. Fortunately, as shown in [50], it can be effectively approximated by considering the subgraph consisting of v_i and its neighboring variables. Specifically, consider the subgraph consisting of v_i and its r-hop neighbors. It has been shown that increasing the diameter of neighborhood (the value of r) can effectively improve the approximation accuracy, and with even a small value of r (e.g. 2-3), r-hop inference can be sufficiently accurate in many real scenarios.

However, in the scenario of gradual inference, some factors (e.g. attribute value similarity) are usually shared by almost all the variables. As a result, r-hop inference may result in a subgraph covering almost all the variables. Therefore, we propose to limit the size of inference subgraph in the following manner: (1) Gradual learning infers the label of a pair based on its features. Approximate inference only needs to consider the factors corresponding to the features of v_i; (2) The influence distribution of a factor is estimated based on its evidence variables. Approximate inference only needs to consider the evidence variables sharing at least one feature with the target inference variable; (3) The total number of evidence variables for any given feature can be limited. As pointed out in [11], the accuracy of function regression generally increases with the number of sample observations. However, the validity of this proposition depends on the uniform distribution of the samples. The additional samples very similar to the existing ones can only produce marginal improvement on prediction accuracy. Therefore, we can limit the total number of evidence variables for each feature by dividing the feature value range of [0,1] into multiple uniform intervals (e.g. 10 intervals, [0,0.1], [0.1,0.2], . . ., [0.9,1.0]), and then limiting the number of observations for each interval (e.g. 200).

It is worthy to point out that our proposed approach for subgraph construction is consistent with the principle of r-hop approximation in that it essentially opts to include those factors and variables in the close neighborhood of a target variable in the subgraph.

7 EMPIRICAL EVALUATION

In this section, we empirically evaluate the performance of GML on real data. We compare GML with both unsupervised and supervised alternatives, which include

- Unsupervised Clustering (denoted by UC). UC maps record pairs to points in a multi-dimensional feature space and then clusters them into distinct classes based on the distance between them. In our implementation, we used the classical k-means to classify pairs into two classes.
- Unsupervised Self-Paced Deep Clustering (denoted by USPDC). We adapt the unsupervised self-paced deep clustering approach proposed for image clustering [10, 21] to ER. Unlike UC, in which instance representation is specified beforehand, USPDC alternates between representation learning and unsupervised clustering. In our implementation, we trained the similarity vector encoder by DeepMatcher [34], which is the state-of-the-art DNN classifier proposed for ER. As in [21], we finetuned a DNN representation model based on self-paced learning and used the classical k-means for clustering.
- Unsupervised Rule-based (denoted by UR). UR reasons about pair equivalence based on the rules handcrafted by the human. Based on knowledge on test data, the rules are specified in terms of record similarity. For fair comparison, in our implementation, UR first uses the result of unsupervised clustering (UC) to estimate the proportions of matching and unmatching instances in a workload, and then proportionally identify the matching and unmatching instances by record similarity.
- Learning based on Support Vector Machine (denoted by SVM). The SVM-based approach [14] also maps record pairs to points in a multi-dimensional feature space. Unlike unsupervised clustering, it fits an optimal SVM classifier on labeled training data and then uses the trained model to label the pairs in test data.
- Deep Learning (denoted by DNN). The deep learning approach [34] is the state-of-the-art supervised learning approach for ER. Representing each record pair by a vector, it first trains a DNN on labeled training data, and then uses the trained model to classify the pairs in test data.

It is noteworthy that the existing semi-supervised learning and active learning techniques are usually applied in the scenario where only a limited number of labeled training data are available. Provided with enough training data, the performance of supervised techniques (e.g. DNN) can be expected to be no worse than their semi-supervised or active learning counterparts. Therefore, the aforementioned four techniques can provide a good coverage of the existing solutions for ER.

7.1 Experimental Setup

Our evaluation is conducted on three real datasets, which are described as follows:

- DBLP-Scholar[3] (denoted by DS): The DS dataset contains the publication entities from DBLP and the publication entities from Google Scholar. The experiments match the DBLP entries with the Scholar entries.
- Abt-Buy[4] (denoted by AB): The AB dataset contains the product entities from both Abt.com and Buy.com. The experiments match the Abt entries with the Buy entries.
- Songs[5] (denoted by SG): The SG dataset contains song entities, some of which refer to the same songs. The experiments match the song entries in the same table.

As in the previous study [34], we use the blocking technique to filter the instance pairs having a small chance to be equivalent. GML computes pair similarity by aggregating the attribute similarities via a weighted sum [15]. For fair comparison, given a percentage of easy instances (e.g. 30%), GML first uses the result of unsupervised clustering (UC) to estimate the proportions of matching and unmatching instances in a workload, and then proportionally identify the easy matching and unmatching instances by record similarity.

We used the platform of *PyTorch* [36] to implement GML. In the comparative study, we set the ratio of easy instances at 30% on all the test workloads. For scalable gradual inference, we set $m = 2000$ and $k = 10$. Our evaluation results in Subsection. 7.3 will show that GML performs very robustly w.r.t various parameter settings. Our implementation codes of GML and the used test datasets have also been made open-source available at the website[6].

7.2 Comparative Study

The detailed evaluation results are presented in Table 2. For SVM and DNN, we report their performance provided with different sizes of training data, which is measured by the fraction of training data among the whole dataset. In Table 2, the percentage of training data is listed at the second low in the table. For instance, for SVM, "30%" means that 30% of a dataset are used for training; for DNN, "30%(25%:5%)" means that 25% of a dataset are used for model training, 5% are used for validation. Since the performance of SVM and DNN depends on the randomly-selected training data, the reported results are the averages over ten runs.

It can be observed that GML performs considerably better than the unsupervised alternatives, UC, USPDC and UR. In most cases, their performance differences in terms of F-1 are larger than 5%. Due to the inherent challenge of ER, the simple UR and UC approaches can not achieve satisfactory performance. It is worthy to point out that the more sophisticated USPDC approach fails to outperform the simpler alternative of UC on the test workloads. On AB, USPDC even performs considerably worse than UC with the margin of more than 0.3. Our closer scrutiny has revealed that even though DeepMatcher provides with a powerful feature representation capability tailored to ER, the "easy" training instances selected by k-means may contain too much label noise. To be more

[3] available at https://dbs.uni-leipzig.de/file/DBLP-Scholar.zip

[4] available at https://dbs.uni-leipzig.de/file/Abt-Buy.zip

[5] available at http://pages.cs.wisc.edu/~anhai/data/falcon_data/songs

[6] https://chenbenben.org/GML/GML-ER.zip

Table 2: Comparative Evaluation of GML

	GML			UC			USPDC			UR		
	recall	precision	F1	recall	precision	F1	recall	precision	F1	recall	precision	F1
DS	0.922	0.927	**0.924**	0.793	0.939	**0.860**	0.920	0.797	**0.854**	0.808	0.958	**0.877**
AB	0.583	0.592	**0.587**	0.689	0.444	**0.540**	0.919	0.130	**0.228**	0.696	0.449	**0.546**
SG	0.982	0.993	**0.987**	0.995	0.808	**0.892**	0.922	0.886	**0.904**	0.994	0.811	**0.893**

	SVM								
	10%			20%			30%		
	recall	precision	F1	recall	precision	F1	recall	precision	F1
DS	0.890	0.918	**0.903**	0.892	0.918	**0.904**	0.896	0.921	**0.908**
AB	0.476	0.677	**0.559**	0.608	0.524	**0.563**	0.676	0.483	**0.563**
SG	0.982	0.992	**0.987**	0.981	0.993	**0.987**	0.980	0.995	**0.987**

	DNN								
	10%(5%:5%)			20%(15%:5%)			30%(25%:5%)		
	recall	precision	F1	recall	precision	F1	recall	precision	F1
DS	0.949	0.869	**0.907**	0.945	0.956	**0.950**	0.982	0.929	**0.955**
AB	0.043	0.254	**0.074**	0.441	0.601	**0.509**	0.444	0.707	**0.546**
SG	0.777	0.830	**0.802**	0.952	0.900	**0.925**	0.938	0.970	**0.954**

specific, USPDC regards the instances closest to a cluster center as the "easy" ones, which are then used for the following iteration of representation learning. In the scenario of ER, this selection strategy may result in noisy training examples. To make matters worse, they may not be able to sufficiently represent the characteristics of more challenging instances, which are further away from the cluster centers. As a result, for USPDC, an initial clustering error can easily snowball after several iterations. Our experimental results clearly illustrate the limitations of USPDC. In contrast, GML labels easy instances only once before gradual inference. Our experimental results have shown that the strategy of considering a pair instance as easy based on record similarity is considerably more accurate than distance-based clustering. Furthermore, as shown in Subsection 7.3, compared with iterative representation learning based on DNN, gradual inference is more robust w.r.t the accuracy of easy instance labeling.

We can also observe that the performance of GML in terms of F-1 is also highly competitive compared to both supervised approaches of SVM and DNN. GML beats both supervised approaches of SVM and DNN in most cases if the percentage of provided training data is no larger than 30%. When the size of training data increases, the performance of SVM and DNN generally improves as expected. Even with the training data size at 30%, GML achieves roughly the same performance as SVM and DNN on all the 3 datasets. *It is worthy to point out that unlike the supervised SVM and DNN models, GML does not use any labeled training data. These experimental results evidently demonstrate the efficacy of GML.*

Table 3: Sensitivity Evaluation w.r.t Easy Instance Labeling

F-1(Easy Acc(%))	30%	40%	50%	80%	100%
DS	0.924(99.7)	0.924(99.7)	0.922(99.3)	0.884(92.3)	0.877(89.6)
AB	0.587(96.5)	0.576(95.3)	0.573(94.4)	0.570(92.2)	0.546(90.0)
SG	0.987(99.7)	0.987(99.6)	0.987(99.5)	0.825(97.3)	0.893(96.0)

Table 4: Sensitivity Evaluation w.r.t the Parameter m

F-1	$m = 500$	$m = 1000$	$m = 2000$
DS	0.922	0.924	0.924
AB	0.587	0.587	0.587
SG	0.987	0.987	0.987

Table 5: Sensitivity Evaluation w.r.t the Parameter k

F-1	$k = 1$	$k = 5$	$k = 10$
DS	0.924	0.924	0.924
AB	0.587	0.588	0.587
SG	0.987	0.987	0.987

Table 6: Sensitivity Evaluation w.r.t the Parameter δ

F-1	$\delta = 50$	$\delta = 100$	$\delta = 200$
DS	0.922	0.924	0.924
AB	0.587	0.583	0.587
SG	0.987	0.987	0.987

7.3 Sensitivity Evaluation

In the sensitivity evaluation, we vary the ratio of the initial easy instances, the number of the pair candidates selected for inference probability approximation (the parameter m in Algorithm 1), and the number of the pair candidates selected for factor graph inference (the parameter k in Algorithm 1). The value of m is set between 500 and 2000, and the value of k is set between 1 and 10. While evaluating the sensitivity of GML w.r.t a specific parameter, we fixed all the other parameters at the same values. The detailed evaluation results are reported in Table 3, 4 and 5.

The evaluation results w.r.t the ratio of easy instances have been shown in Table 3, in which the percentage values in the parentheses represent the accuracy of easy instance labeling. Note that due to the unbalanced numbers of inequivalent and equivalent pairs, the overall high accuracy of easy instance labeling may not necessarily result in similarly high F-1 performance. It can be observed that given a reasonable range on the ratio of easy instances (between 30% and 50%), the performance of GML is very stable. However, it does not mean that GML can afford to set the ratio of easy instances at arbitrarily high. In Table 3, we also report the performance of GML with the ratio set at 80% and 100%. Note that with the ratio of 100%, GML is equivalent to UR. We can observe that in both cases, the performance of GML deteriorates considerably. In GML, the performance of gradual inference depends on the label accuracy of evidential easy instances. If the ratio is set too high, easy instance labeling would introduce considerable errors and the labeling accuracy of hard instances would decrease as well.

Similarly, as shown Table 4 and 5, the performance of GML is highly robust w.r.t the parameters of m and k. Our experimental results bode well for GML's applicability in real applications. It is worthy to point out that even though setting k to a small number can only marginally affect the performance of GML, it does not mean that the factor graph inference is unwanted, can thus be replaced by the more efficient approximate probability estimation. On the contrary, we have observed in the experiments that there actually exist many pair instances whose factor graph inference results are sufficiently different from their approximated probabilities such that their labels are flipped by factor graph inference, especially in the final stages of gradual inference.

7.4 Scalability Evaluation

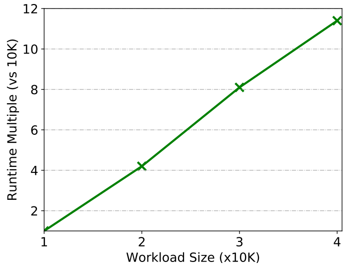


Figure 6: Scalability Evaluation.

In this section, we evaluate the scalability of the proposed scalable approach for GML. Based on the entities in DBLP and Scholar, we generate different-sized DS workloads, from 10000 to 40000. The detailed evaluation results on scalability are presented in Figure 6, in which the x-axis denotes workload size and the y-axis denotes the cost multiple with the runtime spent on the workload of $10k$ as the baseline. It can be observed that the total consumed time increases nearly linearly with workload size. Even though the total number of features consistently increases with workload size, the number of features any instance has is quite stable (in the order of tens). Because the number of evidential observations for each interval of feature values is limited by δ, the average cost of the scalable GML spent on each unlabeled pair only increases marginally as the workload increases. Therefore, the scalable approach scales well with workload size.

8 CONCLUSION

In this paper, we have proposed a novel learning paradigm, called gradual machine learning. We have also developed an effective solution based on it for entity resolution. Finally, our empirical study on real data has validated the efficacy of GML.

Our work on gradual machine learning is an ongoing effort. Using ER as a test case, we have demonstrated that gradual machine learning is a promising paradigm. It is very interesting to develop the solutions based on GML for other challenging classification tasks besides entity resolution and sentiment analysis. On the other hand, even though GML is proposed as an unsupervised learning paradigm in this paper, human work can be potentially integrated into its process for improved performance. An interesting challenge is then how to effectively improve the performance of gradual machine learning with the minimal effort of human intervention, which include but are not limited to manually labeling some instances.

REFERENCES

[1] Ejaz S. Ahmed, Andrei I. Volodin, and Abdulkadir. A. Hussein. Robust weighted likelihood estimation of exponential parameters. *IEEE Transactions on Reliability*, 54(3):389–395, 2005.

[2] Yasser Altowim, Dmitri V. Kalashnikov, and Sharad Mehrotra. Progressive approach to relational entity resolution. *Proceedings of the VLDB Endowment*, 7(11):999–1010, 2014.

[3] Arvind Arasu, Michaela Götz, and Raghav Kaushik. On active learning of record matching packages. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, Indianapolis, Indiana, USA, June 6-10, 2010*, pages 783–794, 2010.

[4] Kedar Bellare, Suresh Iyengar, Aditya G. Parameswaran, and Vibhor Rastogi. Active sampling for entity matching. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 1131–1139, 2012.

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, pages 41–48, 2009.

[6] Mikhail Bilenko, Raymond J. Mooney, William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5):16–23, 2003.

[7] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998.*, pages 92–100, 1998.

[8] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[9] Chengliang Chai, Guoliang Li, Jian Li, Dong Deng, and Jianhua Feng. Cost-effective crowdsourced entity resolution: A partial-order approach. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 969–984, 2016.

[10] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society.

[11] Songxi Chen. Empirical likelihood confidence intervals for linear regression coefficients. *Journal of Multivariate Analysis*, 49(1):24–40, 1994.

[12] Zhaoqiang Chen, Qun Chen, Boyi Hou, Zhanhuai Li, and Guoliang Li. Towards interpretable and learnable risk analysis for entity resolution. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020*, 2020.

[13] Zhiyuan Chen and Bing Liu. *Lifelong Machine Learning, Second Edition*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2018.

[14] Peter Christen. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, pages 151–159, 2008.

[15] Peter Christen. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Springer, 2012.

[16] Xu Chu, John Morcos, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Nan Tang, and Yin Ye. KATARA: A data cleaning system powered by knowledge bases and crowdsourcing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 1247–1261, 2015.

[17] Munir Cochinwala, Verghese Kurien, Gail Lalk, and Dennis Shasha. Efficient data reconciliation. *Information Sciences*, 137(1-4):1–15, 2001.

[18] Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. Reasoning about record matching rules. *Proceedings of the VLDB Endowment*, 2(1):407–418, 2009.

[19] Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[20] Donatella Firmani, Barna Saha, and Divesh Srivastava. Online entity resolution using an oracle. *Proceedings of the VLDB Endowment*, 9(5):384–395, 2016.

[21] X. Guo, X. Liu, E. Zhu, X. Zhu, M. Li, X. Xu, and J. Yin. Adaptive self-paced deep clustering with data augmentation. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[22] Boyi Hou, Qun Chen, Jiquan Shen, Xin Liu, Ping Zhong, Yanyan Wang, Zhaoqiang Chen, and Zhanhuai Li. Gradual machine learning for entity resolution. In *Proceedings of the 2019 The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 3526–3530, 2019.

[23] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pages 200–209, 1999.

[24] Gary King and Langche Zeng. Logistic regression in rare events data. *Political Analysis*, 9(2):137–163, 2001.

[25] Jyrki Kivinen, Alexander J. Smola, and Robert C. Williamson. Online learning with kernels. *IEEE Trans. Signal Processing*, 52(8):2165–2176, 2004.

[26] Pigi Kouki, Jay Pujara, Christopher Marcum, Laura Koehly, and Lise Getoor. Collective entity resolution in familial networks. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 227–236, 2017.

[27] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 1189–1197, 2010.

[28] Guoliang Li, Jiannan Wang, Yudian Zheng, and Michael J. Franklin. Crowdsourced data management: A survey. *IEEE Trans. Knowl. Data Eng.*, 28(9):2296–2319, 2016.

[29] Lingli Li, Jianzhong Li, and Hong Gao. Rule-based method for entity resolution. *IEEE Trans. Knowl. Data Eng.*, 27(1):250–263, 2015.

[30] Rada Mihalcea. Co-training and self-training for word sense disambiguation. In *Proceedings of the Eighth Conference on Computational Natural Language Learning, CoNLL 2004, Held in cooperation with HLT-NAACL 2004, Boston, Massachusetts, USA, May 6-7, 2004*, pages 33–40, 2004.

[31] Chaitanya Mitash, Kostas E. Bekris, and Abdeslam Boularias. A self-supervised learning system for object detection using physics simulation and multi-view pose estimation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 545–551, Sep. 2017.

[32] Alvaro E. Monge and Charles Elkan. The field matching problem: Algorithms and applications. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 267–270, 1996.

[33] Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. *Proceedings of the VLDB Endowment*, 8(2):125–136, 2014.

[34] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 19–34, 2018.

[35] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[37] Alexander J. Ratner, Stephen H. Bach, Henry R. Ehrenberg, and Christopher Ré. Snorkel: Fast training set generation for information extraction. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 1683–1686, 2017.

[38] Sunita Sarawagi and Anuradha Bhamidipaty. Interactive deduplication using active learning. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*, pages 269–278, 2002.

[39] Jeffrey C. Schlimmer and Richard H. Granger. Incremental learning from noisy data. *Machine Learning*, 1(3):317–354, 1986.

[40] Glenn Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.

[41] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[42] Rohit Singh, Vamsi Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. Generating concise entity matching rules. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 1635–1638, 2017.

[43] Parag Singla and Pedro M. Domingos. Entity resolution with markov logic. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 572–582, 2006.

[44] Vasilis Verroios, Hector Garcia-Molina, and Yannis Papakonstantinou. Waldo: An adaptive human interface for crowd entity resolution. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 1133–1148, 2017.

[45] Sibo Wang, Xiaokui Xiao, and Chun-Hee Lee. Crowd-based deduplication: An adaptive approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 1263–1277, 2015.

[46] Yanyan Wang, Qun Chen, Jiquan Shen, Boyi Hou, Murtadha Ahmed, and Zhanhuai Li. Gradual machine learning for aspect-level sentiment analysis. *under review, https://arxiv.org/abs/1906.02502*, 2019.

[47] Steven Euijong Whang, David Marmaros, and Hector Garcia-Molina. Pay-as-you-go entity resolution. *IEEE Trans. Knowl. Data Eng.*, 25(5):1111–1124, 2013.

[48] Jingru Yang, Ju Fan, Zhewei Wei, Guoliang Li, Tongyu Liu, and Xiaoyong Du. Cost-effective data annotation using game-based crowdsourcing. *Proc. VLDB Endow.*, 12(1):57–70, September 2018.

[49] Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014.

[50] Xiaofeng Zhou, Yang Chen, and Daisy Zhe Wang. Archimedesone: Query processing over probabilistic knowledge bases. *Proceedings of the VLDB Endowment*, 9(13):1461–1464, 2016.