Unsupervised Learning of Link Specifications: Deterministic vs. Non-Deterministic

Axel-Cyrille Ngonga Ngomo[1] and Klaus Lyko[1]

Department of Computer Science
University of Leipzig
Johannisgasse 26, 04103 Leipzig
ngonga@informatik.uni-leipzig.de,
WWW home page: http://limes.sf.net

Abstract. Link Discovery has been shown to be of utter importance for the Linked Data Web. In previous works, several supervised approaches have been developed for learning link specifications out of labelled data. Most recently, genetic programming has also been utilized to learn link specifications in an unsupervised fashion by optimizing a parametrized pseudo-F-measure. The questions underlying this evaluation paper are twofold: First, how well do pseudo-F-measures predict the real accuracy of non-deterministic and deterministic approaches across different types of datasets? Second, how do deterministic approaches compare to non-deterministic approaches? To answer these questions, we evaluated linear and Boolean classifiers against classifiers computed by using genetic programming on six different data sets. We also studied the correlation between two different pseudo-F-measures and the real F-measures achieved by the classifiers at hand. Our evaluation suggests that pseudo-F-measures behave differently on the synthetic and real data sets.

1 Introduction

Over the last years, the importance of Link Discovery (LD) as a research topic has increased significantly. This increase was upheld mainly by the ever-growing size of the Linked Data Web and the scalability and accuracy requirements it brings about. The creation of links between knowledge bases, one of the most important steps in the realization of the vision of the Linked Data Web has profited from this boost of research and seen the development of several LD frameworks and approaches [7, 2, 12, 11, 3]. Two main research focuses played a role so far: (1) the determination of time-efficient algorithms [3, 7, 6] for LD and (2) the development of approaches for the efficient computation of link specifications (also called linkage rules) [8, 4, 10, 9]. In most cases, supervised machine learning approaches were used to tackle the second challenge of LD. Approaches developed so far include batch learning using genetic programming [3], the combination of active learning and of linear and Boolean classifiers [8] as well as the combination of active learning and genetic programming [9]. In addition, unsupervised approaches for learning link specifications have been recently developed [10]. While all these approaches have been shown to achieve good results,

unsupervised approaches obviously trump batch and active learning approaches as they do not require any feedback from the user and can still achieve remarkably good performance. In addition, genetic programming approaches yield the central advantage of being able to exploit the whole spectrum of the link specification grammar provided by the framework in which they were implemented. So far, unsupervised approaches to the discovery of link specifications have only been tested with artificially generated benchmark data and low-noise datasets. Moreover, no deterministic approach for the unsupervised discovery of link specifications has been presented so far, although deterministic approaches such as those presented in [8] counterbalance their limitations in expressiveness by being clearly more time-efficient than approaches based on genetic programming.

The aim of this paper is to experimentally examine the unsupervised discovery of link specifications with respect to two main questions:

1. *Are deterministic approaches able to achieve results comparable to those of genetic approaches?* To address this question, we extended the approach presented in [9] and devised an approach for the unsupervised learning of Boolean and linear classifiers which is loosely based on the RAVEN approach [8]. We refrained from reusing the approach presented in [10] as one of the pseudo-measures we rely on was designed especially to work well with this approach. Consequently, using it could have led to a bias in our results.
2. *How well are pseudo-F-measures suited for unsupervised discovery performed on synthetic and real data sets?* Here, we compared the results achieved by the approaches above on the three OAEI 2010 datasets[1] and on three data sets extracted from real data[2]. In addition to the pseudo-F-measure described in [10] (which we dub $\mathcal{F}_u^\beta$), we devised a supplementary pseudo-F-measure $\mathcal{F}_d^\beta$ which relies more on the standard definition of the F_β-measure. We performed a correlation analysis of the values of $\mathcal{F}_u^\beta$, $\mathcal{F}_d^\beta$ and the F_1 measure and detected a surprisingly different behaviour of these measures across our two groups of data sets.

The rest of this paper is structured as follows: We first give an overview of the approaches and measures we used for our experiments. We then present the results of our experimental setup as well as the results of our experiments. For the sake of reproducibility, we chose to use freely available datasets and made the approaches presented herein freely available at the project website.[3] We conclude with a summary of the implications of our results for the LD community.

2 Approaches

In general, a link specification is a classifier C that assigns each element of the set $S \times T$ to one of the classes of $Y = \{+1, -1\}$, where S is called the set of source

[1] Freely available at `http://oaei.ontologymatching.org/2010/`.

[2] Freely available at `http://dbs.uni-leipzig.de/en/research/projects/object_matching/fever/benchmark_datasets_for_entity_resolution`.

[3] `http://saim.sf.net`

instances, while T is the set of target instances. $(s, t) \in S \times T$ is considered by C to be a correct link when $C(s, t) = +1$. Otherwise, (s, t) is considered not be a potential link. We will assume that the classifier C relies on a complex similarity function σ which consists of a combination of atomic similarity measure σ_i. Each of the atomic similarity measures is associated with a parameter ω_i, which is used in main cases as threshold or weight for σ_i. Supervised approaches to the computation of link specifications use labelled training data $L \subseteq S \times T \times Y$ to maximize an objective function such as the distance from the labelled data items to the boundary of the classifier in the case of Support Vector Machines [1]. The idea behind unsupervised approaches to learning link specifications is that they do not to utilize any training data (i.e., $L = \emptyset$). Instead, they aim to optimize an objective function $\mathcal{F}$. In the following, we present the non-deterministic and the deterministic approaches we utilized in our experiments. We then present two different objective functions that are based on the well-know F_β-measure. These functions build the basis for our evaluation.

2.1 Non-Deterministic Approach

Algorithm 1 EAGLE

Require: Sets of instances S and T, size of population, number of iterations
  Get property mapping (S, T)
  Generate initial population
  repeat
    Compute $\mathcal{F}$ for all individuals.
    Apply genetic operators to population
  until Number of iterations is reached
  return Overall fittest individual

The non-deterministic approach we evaluated is based on the EAGLE approach presented in [9] and implemented in the LIMES framework [8]. The approach was modified as described in Algorithm 1. We begin by generating a random population of n individuals. Let G^t be the population at the iteration t. To evolve a population to the generation G^{t+1}, the fitness of each individuals $g^t \in G^t$ is computed. For this purpose, the mapping $M(g^t)$ generated by g^t is evaluated and the value $\mathcal{F}(M(g^t))$ is assigned to g^t. These fitness values build the basis for selecting individuals for the genetic operator *reproduction*. EAGLE uses a tournament setting between two selected individuals to decide which one is copied to the next generation g^{t+1}. On randomly selected individuals the operator *mutation* is applied according to a probability called the *mutation rate*. A mutation can affect an individual in three different ways: First, it can alter the thresholds used by the individual. Second, a mutation can alter the property values that are compared by one of the atomic measures on which the classifier relies. Finally, mutations can modify the measures included in the

individuals. The third genetic operator, *crossover*, operates on two parent individuals and builds a new offspring by swapping two random sub-trees of the parent genotypes. The application of these operators is carried out iteratively until a maximal number of iterations is reached. The result of this process is the mapping M that is returned by the best individual. M is postprocessed as follows: Each $s \in S$ such that $\exists t \in T : (s,t) \in M$ is mapped to $\arg\max_{t \in T} \sigma(s,t)$. The postprocessed mapping is finally returned.

2.2 Deterministic Approaches

Algorithm 2 EUCLID

Require: Specification of the datasets S and T
Require: $\Delta_\omega \in]0,1[$
Require: $\gamma > 1$, $\theta > 0$ with $(\gamma, \theta) \in \mathbb{N}^2$
  Get property mapping (S, T)
  bestClassifier = (1, ..., 1)
  $\Omega = \emptyset$
  for $k = 0 \rightarrow \lfloor 1/\Delta_\omega \rfloor$ **do**
    $\Omega = \Omega \cup \{1 - k\Delta_\omega\}$
  end for
  for $i = 1 \rightarrow n$ **do**
    $\sigma_i^{max} = \arg\max_{\omega \in \Omega, \sigma_i \in \Sigma} \mathcal{F}(\sigma_i \geq \omega)$, $\omega_i^{min} = 0$, $\omega_i^{max} = 1$
  end for
  for $iterations = 1 \rightarrow \theta$ **do**
    $\Delta = \frac{|\omega_i^{max} - \omega_i^{min}|}{\gamma}$
    $C = \arg\max_{i=1}^{n} \mathcal{F}(\omega_i^{min} + k_i \Delta)$
    if $\mathcal{F}(C) > \mathcal{F}$(bestClassifier) **then**
      C = bestClassifier
    else
      bestClassifier = C
    end if
    for $j = 1 \rightarrow n$ **do**
      $\omega_j^{min} = \max(0, \zeta_j)$, $\omega_j^{max} = \min(1, \zeta_j)$
    end for
  end for
  return bestClassifier

Linear and Boolean classifiers have been shown in previous work [8] to also achieve good results on the task of LD. Both types of classifiers can be characterized by a similarity function σ which depends on similarity measures σ_i and parameters ω_i. Linear classifiers $\mathcal{L}$ classify (s,t) as belonging to $+1$ iff $\sum_{i=1}^{n} \omega_i \sigma_i(s,t) \geq 1$. For Boolean classifiers $\mathcal{B}$, the inequality $\bigwedge_{i=1}^{n} \sigma_i(s,t) \geq \omega_i$ must

be fulfilled by the pair (s, t) for it belong to $+1$. In both cases, a classifier can be encoded by the vector $\Omega = (\omega_1, \ldots, \omega_n)$. Determining the best $\mathcal{L}$ or $\mathcal{B}$ would require testing all possible combinations of values $\omega_i \in [0, 1]$, which would be impracticable. The idea behind our algorithm EUCLID (Efficient and Unsupervised Classification for Link Discovery) is to reduce the number of configurations that must be tested by applying a search within the search space whose granularity increases gradually as described in Algorithm 2: We first begin by detecting the right similarity measure for each pair of properties. To achieve this goal, we use the similarity $\sigma_i^{max} = \underset{\omega \in \Omega, \sigma_i \in \Sigma}{\arg\max} \mathcal{F}(\sigma_i \geq \omega)$ for each pair of properties, where $\mathcal{F}(\sigma_i \geq \omega)$ is the value of the pseudo-F-measure (PFM) of the classifier C_0 such that $C_0(s, t) = +1 \Longleftrightarrow \sigma_i(s, t) \geq \omega$, $\Omega = \{\omega > 0 \text{ with } \exists k \in \mathbb{N} : \omega = 1 - k\Delta_\omega\}$ is a set of threshold values and Σ is the set of similarity measures implemented by the framework at hand. Note that Δ_ω is the first of the three parameters required by EUCLID.

In a second step, we compute the actual link specification. Given two parameters $\gamma > 1$ and $\theta > 0$, ω_i^{min} and ω_i^{max} are set to 0 and 1 respectively for each of the similarities σ_i. Then the interval ω_i^{min} and ω_i^{max} is split into γ intervals of the same size $\Delta = (|\omega_i^{max} - \omega_i^{min}|/\gamma$. For each of the possible parametrization $\omega_i = \omega_i^{min} + k\Delta$, $k \in \{0, \ldots, \gamma\}$, EUCLID simply runs all of the resulting classifiers C and compute their fitness $\mathcal{F}(C)$. The current overall best classifier $C = (\zeta_1, \ldots, \zeta_n)$ is used as new reference point. ω_{min}^i is set to $\max\{0, \zeta_i - \Delta\}$ and ω_{max}^i to $\min\{\zeta_i + \Delta, 1\}$ while $\gamma := \gamma/2$. This procedure is repeated θ times and the best overall classifier w.r.t. $\mathcal{F}$ is returned.

3 Pseudo-F-measures

We considered two different PFMs for the automatic discovery of link specifications. The first PFM, dubbed $\mathcal{F}_u^\beta$, was proposed by [10] and is based on the F_β measure. Consequently, it is defined as

$$\mathcal{F}_u^\beta = (1 + \beta^2) \frac{\mathcal{P}_u \mathcal{R}_u}{\beta^2 \mathcal{P}_u + \mathcal{R}_u}. \tag{1}$$

Let $M \subseteq S \times T$ be a mapping generated by an algorithm, S be the set of source instances and T be the set of target instances. $\mathcal{P}_u$ is defined as

$$\mathcal{P}_u(M) = \frac{|\{s | \exists t : (s, t) \in M\}|}{\sum\limits_s |\{t : (s, t) \in M\}|}, \tag{2}$$

while $\mathcal{R}_u$ is computed as follows:

$$\mathcal{R}_u(M) = \frac{|M|}{\min(|S|, |T|)}. \tag{3}$$

Note that $\mathcal{R}_u(M)$ can be larger than 1 as $|M|$ can be larger than $min(|S|, |T|)$. While this does not occur in the setup proposed by [10], it seems rather counterintuitive that a recall measure can lead to values beyond 1. We thus specified

the following novel pseudo-recall dubbed $\mathcal{R}_d$:

$$\mathcal{R}_d(M) = \frac{|\{s|\exists t : (s,t) \in M\}| + |\{t|\exists s : (s,t) \in M\}|}{|S| + |T|}. \quad (4)$$

This pseudo-recall computes the ratio how well all source and target instances are covered by the Mapping M. Thus, $\mathcal{R}_d(M) = 1$ if every $s \in S$ is mapped to at least one $t \in T$ and vice versa. We would argue that it is therewith more in line with the original definition of precision and recall. Our pseudo-F-measure $\mathcal{F}_d$ is thus defined as

$$\mathcal{F}_d^\beta(M) = (1 + \beta^2)\frac{\mathcal{P}_d(M)\mathcal{R}_d(M)}{\beta^2\mathcal{P}_d(M) + \mathcal{R}_d(M)} \text{ with } \mathcal{P}_d = \mathcal{P}_u. \quad (5)$$

4 Experiments and Results

The goal of our experiments was twofold. First, we wanted to know how deterministic approaches perform in comparison to non-deterministic approaches for the discovery of link specifications. The basic intuition here was that if deterministic approaches can achieve F-scores similar to those of non-deterministic approaches, they should be preferred as they are usually more time-efficient. We thus compared the maximal F-measure achieved by each of our approaches on the six different data sets at hand. Moreover, we wanted to measure how well PFM can predict the real performance of classifiers. Especially, we were interested in knowing whether the predictive power of pseudo-F-measures is as reliable on real data as it has been shown to be on synthetic data. Within this context, we were also interested in knowing which setting of β led to the best real F-measure across the different datasets that we used, as $\beta = 0.1$ was suggested in the past [10]. We thus ran our evaluation using both $\mathcal{F}_u$ and $\mathcal{F}_d$ for β-values between 0.1 and 2.0 using a 0.1 increment. We used two different measures to evaluate the correlation between PFM and F_1. In the following, we present the data, algorithmic parameters and correlation measures used for our experiments. We then present our results and discuss their implications for the next steps of research on link discovery.

4.1 Experimental Setup

In all experiments, we assumed that we knew the perfect mapping between the properties. Each experiment was ran on a single thread of an Ubuntu Linux server running JDK1.7 and was allocated maximally 2GB of RAM. The processors were 2.0GHz quadcore Opterons.

Data We ran our experiments on three synthetic and three real datasets. The synthetic datasets consisted of widely used and well-known Persons1, Persons2 and Restaurant datasets from the OAEI2010 set of benchmark data sets. The real

datasets consisted of the ACM-DBLP, Amazon-Google and Abt-Buy datasets that were extracted from websites or databases and for which a gold standard was created manually as reported in [5]. The ACM-DBLP dataset consists of 2,617 source and 2,295 target publications (gold standard: 2,224 links). The Amazon-Google dataset links 1,363 to 3,226 products (gold standard: 1,300 links). Finally, the Abt-Buy dataset links 1,081 to 1,092 products via 1,097 correct links. All non-RDF datasets were transformed into RDF and all attribute values were set to lower case. Apart from this preprocessing, no other preprocessing step was carried out.

Parametrization of the algorithms Four parameters need to be set to run EAGLE: the number of iterations, the size of the population, the crossover rate and the mutation rate. Similarly to [10], we used 20 iterations with a population of 100 individuals. The crossover and mutation rates were set to 0.6. Given that this approach is not deterministic, we ran the experiments 5 times and present the average values in Section 4.2. Note that the standard deviations for the F-measures were always under 5% of the average value. For EUCLID, we used $\Delta_\omega = 0.1$, $\gamma = 4$, $\theta = 10$.

Correlation Measures To measure the accuracy of the algorithms, we used the standard F_1-measure. We were interested in determining whether the pseudo-F-measures $\mathcal{F}_u^\beta$ and $\mathcal{F}_d^\beta$ can be used practically to predict the F_1 measure achieved by an algorithm and which setting of β was the best to achieve this goal. Thus, we measured the correlation of $\mathcal{F}_u^\beta$ and $\mathcal{F}_d^\beta$ with F_1 across different values of β. We used two different correlation measures: The Pearson and the Spearman correlation. We used the *Pearson* correlation to ensure the comparability of our approach with other correlation studies as this correlation is one of the most commonly used. The main drawback of the Pearson correlation is that it is most reliable at detecting linear correlations between distributions. As there was no reason for assuming a linear relationship between our measures, we opted to also use another correlation measure that do not make any assumption upon the type of correlation between the input distributions. We used the *Spearman correlation* [13], which assesses how well a monotonic function can describe the relationship between the input distributions by comparing the ranks of the values in the two input distribution. For both correlations, we used a 2-tailed significance test with a confidence threshold of 95%.

4.2 Results

We first measured the F_1-scores achieved by our approaches when relying on $\mathcal{F}_d$ (see Figure 1) and $\mathcal{F}_u$ (see Figure 2). Our results indicate that EUCLID is in general slightly superior to EAGLE. Note that the linear classifier in combination with $\mathcal{F}_d$ even outperforms the supervised approaches presented in [5] and [9] on the ACM-DBLP data set. In addition the linear model leads to better results than the Boolean model in most cases (expect on the Restaurant dataset). Our

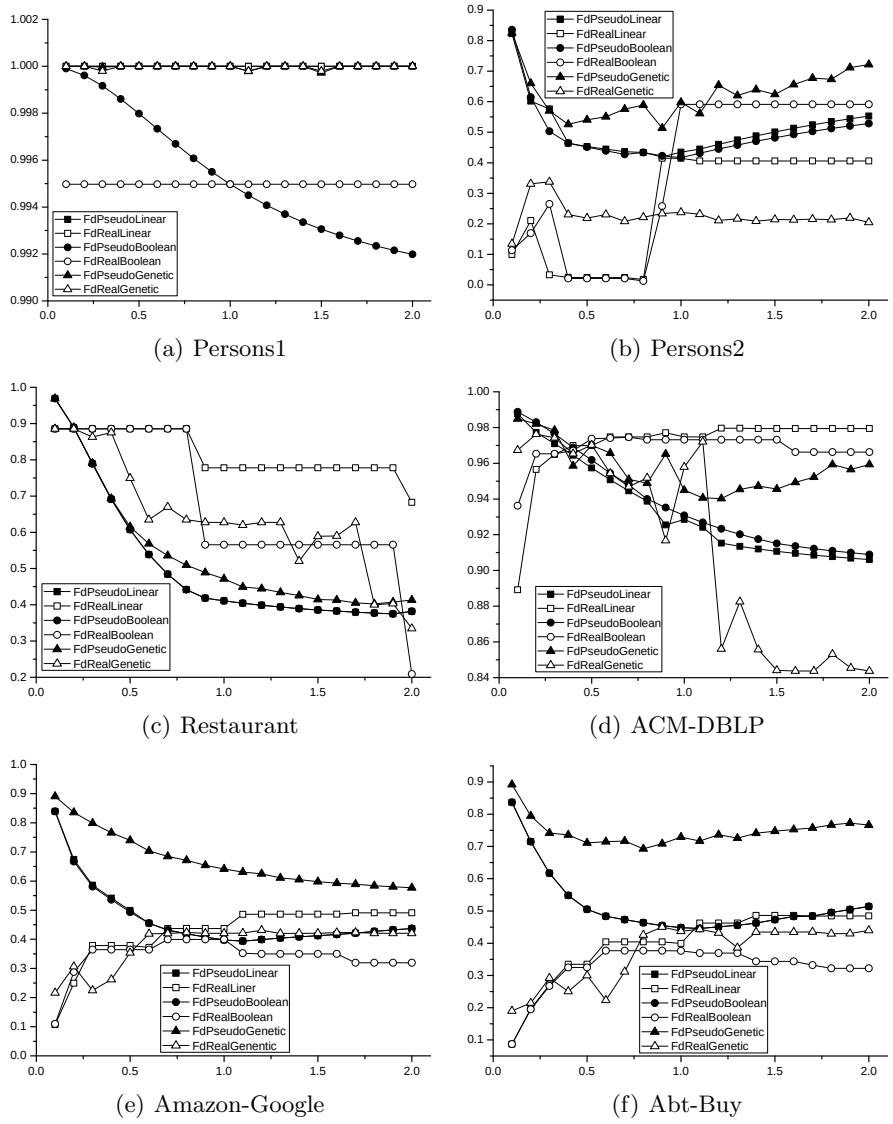
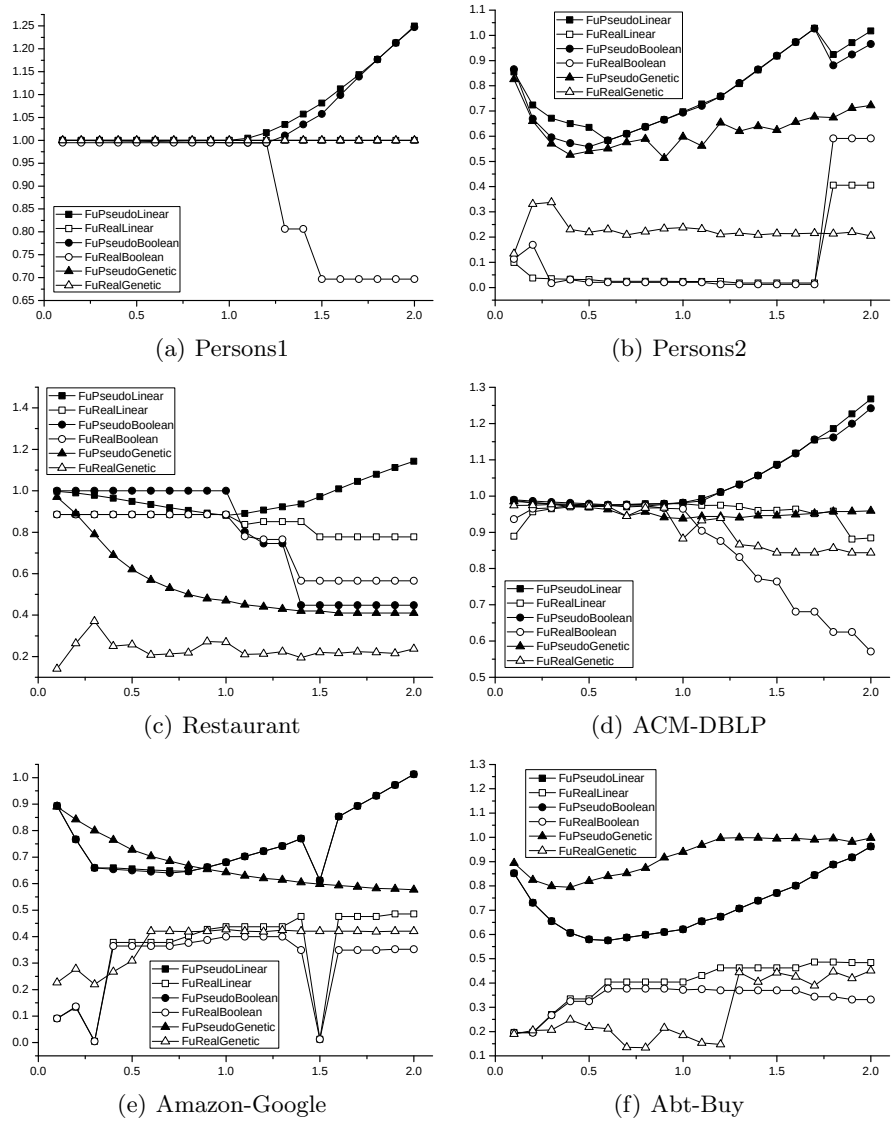


Fig. 1. Evaluation of algorithms based on $\mathcal{F}_d$. The y-axis shows the different F-measures while the x-axis stands for different β-values. Note that "FdPseudo" stands for the pseudo-F-measures achieved by the different classifiers while "FdReal" stands for the real F-measures.

results suggest that $\mathcal{F}_d$ is better suited for EUCLID while $\mathcal{F}_u$ and $\mathcal{F}_d$ tie for EAGLE. With respect to runtime, EUCLID requires between 2.5 and 30s and is therewith between 1 and 2 orders of magnitude faster than EAGLE. Given the significant different in runtimes we observed within our experiments, we suggest



(a) Persons1 (b) Persons2

(c) Restaurant (d) ACM-DBLP

(e) Amazon-Google (f) Abt-Buy

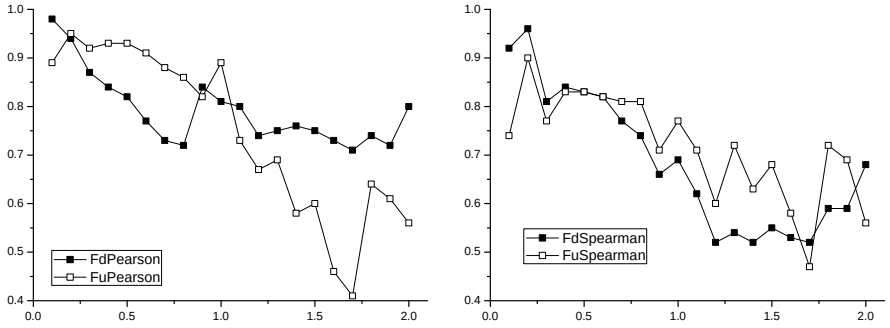
Fig. 2. Evaluation of algorithm based on $\mathcal{F}_u$. The y-axis is the F-measure while the x-axis stands for different β-values. Note that "FuPseudo" stands for the pseudo-F-measures achieved by the different classifiers while "FuReal" stands for the real F-measures.

that the development of specific algorithms for classifiers of a given type can lead to algorithms for the discovery of link specifications that are both time-efficient and highly accurate. The insight we gain is thus a clear *answer to our*

first question: unsupervised deterministic approaches can perform as well as unsupervised approaches on both synthetic and real data.

	Linear		Boolean		Genetic	
	$\mathcal{F}_d$	$\mathcal{F}_u$	$\mathcal{F}_d$	$\mathcal{F}_u$	$\mathcal{F}_d$	$\mathcal{F}_u$
Persons1	**100**	**100**	99.50	99.50	**100**	**100**
Persons2	41.45	40.60	**59.12**	**59.12**	33.77	37.04
Restaurant	**88.56**	**88.56**	**88.56**	**88.56**	**88.56**	**88.56**
ACM-DBLP	**97.96**	97.94	97.46	97.46	97.62	97.71
Amazon-Google	**49.08**	48.55	39.97	39.97	43.11	42.68
Abt-Buy	**48.60**	**48.60**	37.66	37.66	45.03	45.08

Table 1. Maximal F-scores (in %) achieved by the approaches.



(a) Pearson (b) Spearman

Fig. 3. Spearman and Pearson correlation of $\mathcal{F}_u^\beta$ and $\mathcal{F}_d^\beta$ across different values of β measured independently from the algorithm used.

To answer our second question, we first computed the algorithm-independent correlation of $\mathcal{F}_d$ and the F_1 measure as well as the correlation of $\mathcal{F}_u$ and the F_1 measure (see Figure 3). In our experiments, the correlations varied significantly across the different values of β, yet remained positive and significant in 97.5% of the cases (the 2 lowest correlation scores of the Pearson correlation for $\mathcal{F}_u$ were not significant). This means that optimizing $\mathcal{F}_u$ and $\mathcal{F}_d$ for one particular setting of β is a sensible approach towards finding the best classifier for that particular setting of β. We then computed the Pearson and Spearman correlation (see Table 2) between $\mathcal{F}_u$, $\mathcal{F}_d$ and the F_1 measure achieved by the different approaches across different values of β. Our results were somewhat surprising as we detected both significant positive and negative correlations across the different datasets and for both correlations. Interestingly, while the number of

	Linear		Boolean		Genetic	
	$\mathcal{F}_d$	$\mathcal{F}_u$	$\mathcal{F}_d$	$\mathcal{F}_u$	$\mathcal{F}_d$	$\mathcal{F}_u$
Persons1	–	–	–	-0.85*	0.84*	-0.1
Persons2	-0.09	0.54*	-0.12	0.43	-0.43	-0.43
Restaurant	0.73*	-0.71*	0.71*	1*	0.84*	0.16
ACM-DBLP	-0.70*	-0.65*	-0.43	-0.97*	0.46*	0.51*
Amazon-Google	-0.95*	0.49	-0.79*	0.07	-0.88*	-0.03
Abt-Buy	-0.9*	0.27	-0.98*	-0.27	0.38	-0.9*
Persons1	–	–	–	-0.87*	0.99*	-0.1
Persons2	0.02	-0.09	0.21	-0.11	-0.56*	-0.56*
Restaurant	0.85*	-0.54*	0.85*	1*	0.91*	0.15
ACM-DBLP	-0.87*	-0.73*	0.05	-0.95*	0.34	0.47*
Amazon-Google	-0.49*	0.68*	-0.27	-0.22	-0.64*	-0.39
Abt-Buy	-0.37	0.59*	-0.82*	-0.47*	-0.13	-0.49*

Table 2. Pearson and Spearman Correlation of PFM and real F-measures across different β-values. The top section of the table shows the Pearson correlation while the bottom part shows the Spearman correlation. The correlations are not defined for the fields marked with "–" due to at least one of the standard deviations involved being 0. Correlations marked with "*" are significant.

significant positive and negative correlations were relatively balanced for the synthetic data sets, negative correlations seemed to dominate the set of real datasets, thus hinting towards $\mathcal{F}_u$ and $\mathcal{F}_d$ behaving differently depending on the type of data they are confronted with. The negative correlation values suggest that to detect the best values of β for a real dataset automatically, the mapping M which leads to the smallest best value of $\mathcal{F}_d$ across the different values of β should be chosen. This seems rather counter-intuitive and is a hypothesis that requires ampler testing on a larger number of real datasets. Overall, our results show clearly that no β-value achieves a maximal F_1-measure across our data sets. Still, for real datasets, $\mathcal{F}_d$ seems to perform well for $\beta \in [0.8, 1.2]$. Stating such an interval for $\mathcal{F}_u$ is more difficult as the set of β-values that lead to the best mapping is very heterogeneous across the different datasets. Interestingly, this conclusion diverges from that proposed in previous work. The answer to our second question is still clearly that while the predictive power of $\mathcal{F}_u^\beta$ and $\mathcal{F}_d^\beta$ is sufficient for the results to be used in practical settings, significant effort still needs to investigated to create a generic non-parametric PFM that can be used across different datasets and algorithms to predict the F_1-measure reliably .

5 Conclusion

In this paper, we present a first series of experiments to determine how well standard classifier models such as linear and Boolean classifiers perform in com-

parison to classifiers generated by the means of genetic programming in an unsupervised learning setting based on maximizing a PFM. Overall, our results indicate that we are still at the beginning of the search towards the "holy grail" of PFMs. Especially on real data, the maximal PFM achieved by algorithms across different values of β is often negatively correlated with the value of the F_1. The magnitude of this effect is significantly reduced on synthetic data. This difference suggest that there is still a need for benchmark generation methods that allow creating benchmark data sets which reflect real data in a more holistic way. Moreover, our evaluation shows that deterministic classifiers perform as well as or better than non-deterministic approaches while still bearing the main advantage of being significantly more time-efficient. Thus, finding more efficient extension of EUCLID or similar approaches should allow providing users of LD frameworks with accurate link specifications within an interactive setting. Detecting the right parametrization for PFM yet remains an unsolved problem.

References

1. Nello Cristianini and Elisa Ricci. Support vector machines. In *Encyclopedia of Algorithms*. 2008.
2. Aidan Hogan, Axel Polleres, Jrgen Umbrich, and Antoine Zimmermann. Some entities are more equal than others: statistical methods to consolidate linked data. In *Workshop on New Forms of Reasoning for the Semantic Web: Scalable & Dynamic (NeFoRS2010)*, 2010.
3. R. Isele, A. Jentzsch, and C. Bizer. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In *WebDB*, 2011.
4. Robert Isele and Christian Bizer. Learning Linkage Rules using Genetic Programming. In *Sixth International Ontology Matching Workshop*, 2011.
5. Hanna Köpcke, Andreas Thor, and Erhard Rahm. Comparative evaluation of entity resolution approaches with fever. *Proc. VLDB Endow.*, 2(2):1574–1577, 2009.
6. Axel-Cyrille Ngonga Ngomo. A time-efficient hybrid approach to link discovery. In *Proceedings of OM@ISWC*, 2011.
7. Axel-Cyrille Ngonga Ngomo and Sören Auer. Limes - a time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of IJCAI*, 2011.
8. Axel-Cyrille Ngonga Ngomo, Jens Lehmann, Sören Auer, and Konrad Höffner. Raven: Active learning of link specifications. In *Proceedings of the Ontology Matching Workshop (co-located with ISWC)*, 2011.
9. Axel-Cyrille Ngonga Ngomo and Klaus Lyko. Eagle: Efficient active learning of link specifications using genetic programming. In *Proceedings of ESWC*, 2012.
10. Andriy Nikolov, Mathieu D'Aquin, and Enrico Motta. Unsupervised learning of data linking configuration. In *Proceedings of ESWC*, 2012.
11. George Papadakis, Ekaterini Ioannou, Claudia Niedere, Themis Palpanasz, and Wolfgang Nejdl. Eliminating the redundancy in blocking-based entity resolution methods. In *JCDL*, 2011.
12. Jennifer Sleeman and Tim Finin. Computing foaf co-reference relations with rules and machine learning. In *Proceedings of the Third International Workshop on Social Data on the Web*, 2010.
13. C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15:72–101, 1904.