A Hierarchical Clustering Method for Semantic Knowledge Bases

Nicola Fanizzi and Claudia d'Amato

Dipartimento di Informatica, Università degli Studi di Bari
Campus Universitario, Via Orabona 4, 70125 Bari, Italy
{fanizzi|claudia.damato}@di.uniba.it

Abstract. This work presents a clustering method which can be applied to relational knowledge bases. Namely, it can be used to discover interesting groupings of semantically annotated resources in a wide range of concept languages. The method exploits a novel dissimilarity measure that is based on the resource semantics w.r.t. a number of dimensions corresponding to a committee of features, represented by a group of concept descriptions (discriminating features). The algorithm is an adaptation of the classic Bisecting k-Means to complex representations typical of the ontology in the Semantic Web. We discuss its complexity and the potential applications to a variety of important tasks.

Key words: Description Logics, Hierarchical Clustering Algorithm, Medoids, Semantic Web

1 Introduction

In the inherently distributed applications related to the Semantic Web (henceforth SW) there is an extreme need of automatizing those activities which are more burdensome for the knowledge engineer, such as ontology construction, matching and evolution. An automatization of these activities may be achieved through the implementation of supervised or unsupervised inductive methods. In this work, we investigate on unsupervised learning for knowledge bases expressed in the standard ontological languages. In particular, we focus on conceptual clustering of semantically annotated resources.

Essentially, clustering methods are based on the application of similarity (or density) measures, defined over a fixed set of attributes of the domain objects, with the goal of creating classes, namely homogeneous data subgroups. Classes of objects are taken as collections that exhibit low interclass similarity (density) and high intraclass similarity (density). Often these methods cannot take into account any form of *background knowledge* that could characterize object configurations by means of global concepts and semantic relationship. This hinders the interpretation of the outcomes of these methods which, on the contrary, is crucial in the SW perspective that foresees sharing and reusing the produced knowledge in order to enable semantic interoperability. Alternative approaches, particularly suitable for concept languages and terminological representations,

have pursued a different way for attacking the problem, devising logic-based methods [7, 4]. Yet it has been pointed out that these methods may suffer from noise in the data.

This motivates our investigation on similarity-based clustering methods which may be more noise-tolerant, still saving the advantages of conceptual clustering. We propose an extension of effective clustering techniques to a multi-relational setting. Specifically, our relational method derives from the *Bisecting k-means* algorithm [5], a well-known partitional clustering method, and it is tailored for the SW context (see Sect.3). It is intended for grouping similar resources w.r.t. a semantic dissimilarity measure which allows for discovering new concepts. As for the original method, one may fix a given number k of clusters of interest, yet this may be hard when scarce knowledge about the domain is available. As an alternative, a partitional method like ours may be employed up to reaching a minimal threshold value for cluster quality (many measures have been proposed in the literature [5]) which makes any further bisection useless. Moreover, instead of the notion of means, that characterizes the algorithms descending from *k-means* and EM [5] developed for numeric (or just ordinal) features, in our case we recur to the notion of *medoids* (like in algorithm PAM [6] or CLARANS [8]) as central individuals in a cluster.

From a technical viewpoint, upgrading existing algorithms to work on multi-relational representations, like the concept languages used in the SW, requires novel similarity measures that are suitable for such representations. The notion of similarity to be employed has to deal with the rich representations of semantically annotated resources. Therefore, we developed a measure which could be used specifically for the SW standard representations (see below).

As pointed out in a seminal paper [2] on similarity measures for *Description Logics* (DLs), most of the existing measures focus on the similarity of atomic concepts within hierarchies or simple ontologies. Hence, they have been conceived for assessing *concept* similarity. Nevertheless, for our purposes, a notion of similarity between *individuals* is required.

Recently, dissimilarity measures for specific DLs have been proposed [3]. Although they turned out to be quite effective for the inductive tasks, they are still partly based on structural criteria which determine their main weakness: they are hardly scalable to deal with standard languages used in the current knowledge management frameworks. Therefore, we have devised a new semantic dissimilarity measure for semantically annotated resources (see Sect.2), which can overcome the aforementioned limitations.

Following some ideas introduced in [9], we present a new family of measures that is suitable for a wide range of ontology languages (RDF through OWL) since it is merely based on the discernibility of the input individuals with respect to a fixed set of features represented by concept definitions. As such, the new measures are not absolute, yet they depend on the knowledge base they are applied to.

2 Concept Similarity and Semantic Distance Measures

One of the strong points of our method is that it does not rely on a particular language for semantic annotations. Hence, in the following, we assume that resources, concepts and their relationship may be defined in terms of a generic ontology language that may be mapped to some description logic with the standard model-theoretic semantics (see the handbook [1] for a thorough reference). In order to exploit the method, the underlying application needs only to model the knowledge base according to this ontology language. Thus, for instance, a SW service may be searched in a registry, provided that services are described in the standard representations defined on top of OWL (e.g. OWL-S or WSML).

In the SW context, a *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* $\mathcal{T}$ and an *ABox* $\mathcal{A}$. $\mathcal{T}$ is a set of concept definitions. $\mathcal{A}$ contains assertions (facts, data) concerning the world state. Normally the *unique names assumption* is made on the individuals[1] in the ABox. The set of the individuals occurring in $\mathcal{A}$ will be denoted by $\mathsf{Ind}(\mathcal{A})$.

As regards the inference services, like all other instance-based methods, our procedure may require performing *instance-checking*, which amounts to determining whether an individual, say a, belongs to a concept extension, i.e. whether $C(a)$ holds for a certain concept C.

Since the main goal of the proposed method is to make clusters of individuals (asserted in an ontology), for our purposes, we need of a function for measuring the similarity of individuals rather than concepts. Anyway individuals do not have a syntactic structure that can be compared. This has led to lifting them to the concept description level before comparing them (recurring to the approximation of the *most specific concept* of an individual w.r.t. the ABox). Hence, for the clustering procedure specified in Sect. 3, we have developed a new measure whose definition totally depends on semantic aspects of the individuals in the knowledge base.

2.1 The Measure

On a semantic level, similar individuals should behave similarly with respect to the same concepts. On the ground of such an intuition, we introduce a novel measure for assessing the similarity of individuals in a knowledge base, which is based on comparing their semantics along a number of dimensions represented by a committee of concept descriptions. Following the ideas borrowed from ILP [9] and *multi-dimensional scaling*, we propose the definition of totally semantic distance measures for individuals in the context of a knowledge base.

The rationale of the new measure is to compare individuals on the grounds of their behavior w.r.t. a given set of hypotheses, that is a collection of concept descriptions, say $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, which stands as a group of discriminating *features* expressed in the language taken into account.

In its simple formulation, a family of distance functions for individuals inspired to Minkowski's distances can be defined as follows:

[1] Individuals can be assumed to identified by their own URI.

Definition 1 (family of semi-distance measures). *Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base. Given set of concept descriptions $\mathsf{F} = \{F_1, F_2, \ldots, F_m\}$, a family of semi-distance functions $d_p^{\mathsf{F}} : \mathsf{Ind}(\mathcal{A}) \times \mathsf{Ind}(\mathcal{A}) \mapsto \mathbb{R}$ defined as follows:*

$$\forall a, b \in \mathsf{Ind}(\mathcal{A}) \quad d_p^{\mathsf{F}}(a, b) := \frac{1}{m} \left[\sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(b) \mid^p \right]^{1/p}$$

where $p > 0$ and $\forall i \in \{1, \ldots, m\}$ the projection function *π_i is defined by:*

$$\forall a \in \mathsf{Ind}(\mathcal{A}) \quad \pi_i(a) = \begin{cases} 1 & F_i(x) \in \mathcal{A} \\ 0 & \neg F_i(x) \in \mathcal{A} \\ 1/2 & \textit{otherwise} \end{cases}$$

The superscript F will be omitted when the set of hypotheses is fixed.

As an alternative, the definition of the measures can be made more accurate by considering entailment rather than the simple ABox look-up, when determining the values of the projection functions:

$$\forall a \in \mathsf{Ind}(\mathcal{A}) \quad \pi_i(a) = \begin{cases} 1 & \mathcal{K} \models F_i(x) \\ 0 & \mathcal{K} \models \neg F_i(x) \\ 1/2 & \textit{otherwise} \end{cases}$$

In particular, for the sake of saving computational resources, we have considered the following measures in the experiments: $\forall a, b \in \mathsf{Ind}(\mathcal{A})$

$$d_1(a, b) := \frac{1}{m} \sum_{i=1}^{m} \mid \pi_i(a) - \pi_i(b) \mid \quad \text{or} \quad d_2(a, b) := \frac{1}{m} \sqrt{\sum_{i=1}^{m} (\pi_i(a) - \pi_i(b))^2}$$

2.2 Discussion

It is easy to prove that these functions have the standard properties for semi-distances:

Proposition 1 (semi-distance). *For a fixed feature set and $p > 0$, given any three instances $a, b, c \in \mathsf{Ind}(\mathcal{A})$. it holds that:*

1. $d_p(a, b) > 0$
2. $d_p(a, b) = d_p(b, a)$
3. $d_p(a, c) \leq d_p(a, b) + d_p(b, c)$

It cannot be proved that $d_p(a, b) = 0$ iff $a = b$. This is the case of *indiscernible* individuals with respect to the given set of hypotheses F.

Compared to other proposed distance (or dissimilarity) measures [2], the presented function does not depend on the constructors of a specific language, rather it requires only retrieval or instance-checking service used for deciding whether an individual is asserted in the knowledge base to belong to a concept extension (or, alternatively, if this could be derived as a logical consequence).

Note that the π_i functions ($\forall i = 1, \ldots, m$) for the training instances, that contribute to determine the measure with respect to new ones, can be computed in advance thus determining a speed-up in the actual computation of the measure. This is very important for the measure integration in algorithms which massively use this distance, such as all instance-based methods.

The underlying idea for the measure is that similar individuals should exhibit the same behavior w.r.t. the concepts in F. Here, we make the assumption that the feature-set F represents a sufficient number of (possibly redundant) features that are able to discriminate really different individuals. The choice of the concepts to be included – *feature selection* – is beyond the scope of this work. Experimentally, we could obtain good results by using the very set of both primitive and defined concepts found in the ontology.

3 Hierarchical Clustering around Medoids

The conceptual clustering procedure implemented in our method works top-down, starting with one universal cluster grouping all instances. Then it iteratively finds two clusters bisecting an existing one up to the desired number of clusters is reached. This algorithm can be thought as producing a dendrogram levelwise: the number of levels coincides with the number of clusters. It can be very fast.

In particular our algorithm can be ascribed to the category of the heuristic partitioning algorithms such as K-MEANS and EM [5]. Each cluster is represented by the center of the cluster. In our setting we will consider the notion of medoid as a notion of cluster center since our distance measure works on a categorical feature-space. In particular it can be seen as a hierarchical extension of the PAM algorithm (*Partition Around Medoids* [6]): each cluster is represented by one of the individuals in the cluster, the medoid, i.e., in our case, the one with the lowest average distance w.r.t. all the others individuals in the cluster. The bi-partition is repeated level-wise producing a dendrogram.

Fig. 1 reports a sketch of our algorithm. It essentially consists of two nested loops: the outer one computes a new level of the resulting dendrogram and it is repeated until the desired number of clusters is obtained (which corresponds to the latest level; the inner loop consists of a run of the PAM algorithm at the current level.

Per each level, the next worst cluster is selected (*selectWorstCluster*() function) on the grounds of its quality, e.g. the one endowed with the least average inner similarity (or cohesiveness [10]). This cluster is candidate to being parted in two. The partition is constructed around two medoids initially chosen (*selectMostDissimilar*() function) as the most dissimilar elements in the cluster and then iteratively adjusted in the inner loop. In the end, the candidate cluster is replaced by the newly found parts at the next level of the dendrogram.

The inner loop basically resembles to a 2-means (or EM) algorithm, where medoids are considered instead of means which can hardly be defined in symbolic computations. Then, the classical two steps are performed in an iteration:

```
clusterVector HierarchicalBisectingAroundMedoids(allIndividuals, k, maxIterations)
input  allIndividuals: set of individuals
       k: number of clusters;
       maxIterations: max number of inner iterations;
output clusterVector: array [1..k] of sets of clusters

begin
level := 0;
clusterVector[1] := allIndividuals;
repeat
       ++level;
       cluster2split := selectWorstCluster(clusterVector[level]);
       iterCount := 0;
       stableConfiguration := false;
       (newMedoid1,newMedoid2) := selectMostDissimilar(cluster2split);
       repeat
            ++iterCount;
            // E step
            (medoid1,medoid2) := (newMedoid1,newMedoid2);
            (cluster1,cluster2) := distribute(cluster2split,medoid1,medoid2);
            // M step
            newMedoid1 := medoid(cluster1);
            newMedoid2 := medoid(cluster2);
            stableConfiguration := (medoid1 = newMedoid1) ∧ (medoid2 = newMedoid2);
       until stableConfiguration ∨ (iterCount = maxIterations);
       clusterVector[level+1] := replace(cluster2split,cluster1,cluster2,clusterVector[level]);
until (level = k);
end
```

Fig. 1. The Hierarchical Bisecting around Medoids Algorithm.

E step given the current medoids, the first distributes the other individuals in one of the two partitions under construction on the grounds of their similarity w.r.t. either medoid;

M step given the bipartition obtained by *distribute*(), this second step computes the new medoids for either cluster. These tend to change on each iteration until eventually they converge to a stable couple (or when a maximum number of iteration have been performed).

The medoid of a group of individuals is the individual that has the lowest distance w.r.t. the others. Formally. given a cluster $C = \{a_1, a_2, \ldots, a_n\}$, the medoid is defined:

$$m = \text{medoid}(C) = \operatorname*{argmin}_{a \in C} \sum_{j=1}^{n} d(a, a_j)$$

Each node of the tree (a cluster) may be labeled with an intensional concept definition which characterizes the individuals in the given cluster while discriminating those in the twin cluster at the same level. Labeling the tree-nodes with

concepts can be regarded as a number of supervised learning problems in the specific multi-relational representation targeted in our setting. As such it deserves specific solutions that are suitable for the DL languages employed.

A straightforward solution may be found, for DLs that allow for the computation of (an approximation of) the *most specific concept* (msc) and *least common subsumer* (lcs) [1] (such as $\mathcal{ALC}$). This may involve the following steps:
given a cluster of individuals node_j

- **for each** individual $a_i \in \mathsf{node}_j$ **do**
  compute $M_i := \mathsf{msc}(a_i)$ w.r.t. $\mathcal{A}$;
- **let** $\mathsf{MSCs}_j := \{M_i \mid a_i \in \mathsf{node}_j\}$;
- **return** $\mathsf{lcs}(\mathsf{MSCs}_j)$

As an alternative, algorithms for learning concept descriptions expressed in DLs may be employed [4]. Indeed, concept formation can be cast as a supervised learning problem: once the two clusters at a certain level have been found, the members of a cluster are considered as positive examples and the members of the dual cluster as negative ones. Then any concept learning method which can deal with this representation may be utilized for this new task.

The representation of centers by means of medoids has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers. In k-means case, a cluster is represented by its centroid, which is a mean (usually weighted average) of points within a cluster. This works conveniently only with numerical attributes and can be negatively affected by a single outlier.

A PAM algorithm has several favorable properties. Since it performs clustering with respect to any specified metric, it allows a flexible definition of similarity. This flexibility is particularly important in biological applications where researchers may be interested, for example, in grouping correlated or possibly also anti-correlated elements. Many clustering algorithms do not allow for a flexible definition of similarity, but allow only Euclidean distance in current implementations. In addition to allowing a flexible distance metric, a PAM algorithm has the advantage of identifying clusters by the medoids. Medoids are robust representations of the cluster centers that are less sensitive to outliers than other cluster profiles, such as the cluster means of K-MEANS. This robustness is particularly important in the common context that many elements do not belong exactly to any cluster, which may be the case of the membership in DL knowledge bases, which may be not ascertained given the OWA.

The benefits of *conceptual clustering* [10] in the context of semantically annotated knowledge bases are manifold:

- *concept formation*: clustering annotated resources enables the definition of new emerging concepts on the grounds of the primitive concepts asserted in a knowledge base;
- *evolution*: supervised methods can exploit these clusters to induce new concept definitions or to refining existing ones;

– *discovery and ranking*: intensionally defined groupings may speed-up the task of search and discovery; a hierarchical clustering suggests criteria for ranking the retrieved resources.

4 Conclusions

This work has presented a clustering for (multi-)relational representations which are standard in the SW field. Namely, it can be used to discover interesting groupings of semantically annotated resources in a wide range of concept languages. The method exploits a novel dissimilarity measure, that is based on the resource semantics w.r.t. a number of dimensions corresponding to a committee of features represented by a group of concept descriptions (discriminating features). The algorithm, is an extension of the classic bisecting k-means to complex representations typical of the SW ontology languages. We have discussed its complexity and the potential applications to a variety of important tasks.

References

1. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
2. A. Borgida, T.J. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In I. Horrocks, U. Sattler, and F. Wolter, editors, *Working Notes of the International Description Logics Workshop*, volume 147 of *CEUR Workshop Proceedings*, Edinburgh, UK, 2005.
3. C. d'Amato, N. Fanizzi, and F. Esposito. Reasoning by analogy in description logics through instance-based learning. In G. Tummarello, P. Bouquet, and O. Signore, editors, *Proceedings of Semantic Web Applications and Perspectives, 3rd Italian Semantic Web Workshop, SWAP2006*, volume 201 of *CEUR Workshop Proceedings*, Pisa, Italy, 2006.
4. N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Concept formation in expressive description logics. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *Proceedings of the 15th European Conference on Machine Learning, ECML2004*, volume 3201 of *LNAI*, pages 99–113. Springer, 2004.
5. A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
6. L. Kaufman and Rousseeuw. P.J. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.
7. J.-U. Kietz and K. Morik. A polynomial approach to the constructive induction of structural knowledge. *Machine Learning*, 14(2):193–218, 1994.
8. R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proceedings of the 20th Conference on Very Large Databases, VLDB94*, pages 144–155, 1994.
9. M. Sebag. Distance induction in first order logic. In S. Džeroski and N. Lavrač, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming, ILP97*, volume 1297 of *LNAI*, pages 264–272. Springer, 1997.
10. R. E. Stepp and R. S. Michalski. Conceptual clustering of structured objects: A goal-oriented approach. *Artificial Intelligence*, 28(1):43–69, Feb. 1986.