Journal of Artificial Intelligence Research 63 (2018) 691-742 Submitted 02/18; published 11/18

Data-Driven Conceptual Spaces: Creating Semantic Representations for Linguistic Descriptions of Numerical Data

Hadi Banaee HADI.BANAEE@ORU.SE
Erik Schaffernicht ERIK.SCHAFFERNICHT@ORU.SE
Amy Loutfi AMY.LOUTFI@ORU.SE
Centre for Applied Autonomous Sensor Systems
School of Science and Technology
Örebro University, SE-701 82 Örebro, Sweden

Abstract

There is an increasing need to derive semantics from real-world observations to facilitate natural information sharing between machine and human. Conceptual spaces theory is a possible approach and has been proposed as mid-level representation between symbolic and sub-symbolic representations, whereby concepts are represented in a geometrical space that is characterised by a number of quality dimensions. Currently, much of the work has demonstrated how conceptual spaces are created in a knowledge-driven manner, relying on prior knowledge to form concepts and identify quality dimensions. This paper presents a method to create semantic representations using data-driven conceptual spaces which are then used to derive linguistic descriptions of numerical data. Our contribution is a principled approach to automatically construct a conceptual space from a set of known observations wherein the quality dimensions and domains are not known a priori. This novelty of the approach is the ability to select and group semantic features to discriminate between concepts in a data-driven manner while preserving the semantic interpretation that is needed to infer linguistic descriptions for interaction with humans. Two data sets representing leaf images and time series signals are used to evaluate the method. An empirical evaluation for each case study assesses how well linguistic descriptions generated from the conceptual spaces identify unknown observations. Furthermore, comparisons are made with descriptions derived on alternative approaches for generating semantic models.

1. Introduction

Artificial intelligence systems that interact with humans in natural language must be able to form concepts either directly from observations or indirectly from other concepts. New applications within several domains, such as cognitive robotics, will increasingly need to rely on automatic methods to form concepts directly from sensor data. Different approaches to creating the representational models necessary for *concept formation* have been described in the literature of artificial intelligence (AI). *Symbolic* approaches use explicit symbols as primitives to perform symbol manipulation in order to model high-level abstract concepts (Sun, 1999; Minsky, 1991). *Sub-symbolic* approaches focus often on the categorisation tasks per se, and process the activation patterns of concepts in the perceptual level using internally connected units (Sun, 1999; Gärdenfors, 1997). Neither approach is fully satisfactory (Gärdenfors, 1995). Generally, symbolic approaches do not perform the task of *inductive* *inference* or *concept learning*, which is the process of performing a generalisation from a limited number of observations (Sun, 1999). On the other hand, sub-symbolic approaches often neglect the issue of *explainability* or *interpretability*, that is, the process of inferring meaningful descriptions from a set of (semantically enriched) representations. For instance, *connectionist* approaches are often not able to explain *what* the emerging learnt model represents (Aisbett & Gibbon, 2001).

Consequently, the theory of conceptual spaces was introduced by Gärdenfors (2000) as a mid-level representation between symbolic and sub-symbolic approaches (Aisbett & Gibbon, 2001) to address both concept learning and semantic inference (Aisbett & Gibbon, 2001; Holender, Nagi, Sudit, & Rickard, 2007). Conceptual spaces theory presents a framework consisting of a set of *quality dimensions* (i.e. cognitively meaningful attributes) in various *domains* within a geometrical structure in order to model, categorise, and represent the concepts in a multi-dimensional space (Gärdenfors, 2000). Concepts in a conceptual space are represented by their properties as regions within the domains. Criticism on the theory of conceptual spaces, an open question is how to automatically construct a conceptual space from numerical data (Keßler, 2006). In the literature, conceptual spaces have been principally derived in a *knowledge-driven* manner and operate on the assumption that there is prior knowledge from perceptual mechanisms or experts to manually initialise the elements of the conceptual space (i.e., domains, quality dimensions, and concepts' regions) (Agostaro, Augello, Pilato, Vassallo, & Gaglio, 2005; Rickard, Aisbett, & Gibbon, 2007).

This paper presents a *data-driven* approach to automatically *construct* conceptual spaces and perform concept formation based on input observations (i.e., exemplars) in a numerical data set. Instead of initialising domains and dimensions from a priori knowledge or forming concepts in a rule-based manner, the proposed data-driven construction process automatically determines the relevant domains and dimensions based on the ability to distinguish between exemplars of different concepts. We further propose a semantic inference process for the built conceptual space to provide explainability of new and unknown observations in natural language descriptions.

The remainder of the paper is structured as follows. Section 2 provides background on conceptual spaces and related work. In Section 3 we introduce our approach of automatically constructing conceptual spaces from observations. Afterwards, we present the process of inferring semantic descriptions for a set of unknown observation via the constructed conceptual space in Section 4. In Section 5 we present two case studies where our proposed approach is exploited. The output descriptions from the case studies are then assessed in Section 6 based on an empirical evaluation designed to show the advantages of our proposed framework. Finally, in Sections 7 and 8 we discuss the method and the obtained results and draw our conclusion.

2. Background and Related Work

In this section, we provide background on the theory of conceptual spaces. We then discuss related work focusing on two areas: First, we describe the role of the conceptual spaces theory in the field of AI, and the necessity of revising its definitions based on AI needs. Second, we present a brief review of the relevant semantic inference and linguistic description approaches, followed by a discussion of the advantage of using conceptual spaces as a semantic representation.

2.1 On the Theory of Conceptual Spaces

The theory of conceptual spaces was introduced as a knowledge representation framework which relies on the paradigm of cognitive semantics (Gärdenfors, 2000). This theory explores how various types of information can be represented, both from a psychological perspective and for developing an artificial system (Gärdenfors, 2004). A conceptual space is a geometrical structure which is defined by a set of *quality dimensions*. The quality dimensions present the features of objects in the space based on their measured quality values. One conceptual space can consist of multiple domains. A *domain* in the conceptual space is represented as a set of interdependent quality dimensions which are logically integrated. A typical example of a domain is *colour* that is defined by the quality dimensions *hue*, *saturation*, and *brightness*. Other examples of perceptual domains are *shape*, *size*, and *weight* (Gärdenfors, 2000).

Concepts in a conceptual space are represented as a set of regions through multiple domains. A concept is described based on its different properties in various domains. *Properties* are the convex regions in a single domain expressing a particular attribute of the domain. For example, *green* is a property corresponding to a region in the *colour* domain. An *instance* of a concept is defined as a set of points in the conceptual space, in which each point is located within a property region of that concept in one of the domain. As an example, in a conceptual space of *fruits* including the domains *colour*, *taste*, and *size*, the concept of *apple* can be represented as a set of regions within these domains (LeBlanc, 2010). In this conceptual space, one instance (object) of the concept of *apple* can take place in the properties of *green*, *medium*, and *sweet*, which are geometrical regions within the domains *colour*, *size*, and *taste*, respectively.

The metric definition of domains allows depicting the notion of *similarity* in a conceptual space. Measuring the similarity robustly eases the consideration of cognitive tasks such as concept formation, semantic inferences, induction, and concept learning (Holender et al., 2007). In concept representation, it is possible to assign weights to the domains or dimensions in order to distinguish between similar concepts in different contexts (Gärdenfors, 2004), which includes the notion of *context* into the conceptual spaces theory. Various formalisations of the conceptual spaces theory have been proposed in the literature (Aisbett & Gibbon, 2001; Rickard et al., 2007; Raubal, 2004; Holender et al., 2007), which attempted to mathematically formalise how to construct and perform induction in conceptual spaces.

2.2 Related Work on Conceptual Spaces and AI

As mentioned in the introduction, from the AI point of view, the aim of representing knowledge in a conceptual space is to develop an intuitive interpretation of the relationship between symbolic and sub-symbolic information (Gärdenfors, 2000; Aisbett & Gibbon, 2001). Gärdenfors has discussed thoroughly the role of conceptual spaces as a knowledge representation framework in AI systems (Gärdenfors, 2004), focusing on the tasks of *induction* and *reasoning* (Gärdenfors & Williams, 2001; Gärdenfors, 2005). Recently, Lieto et. al. (2017) have detailed the need for a conceptual representation as a knowledge representation level in-between the symbolic and the sub-symbolic one. This offers cognitive architectures a common language enabling the interaction between different types of representations. Schockaert and Prade (2013) have focused on the problem of interpolative and extrapolative inference for different properties and concepts with the help of conceptual spaces. In addition to the theoretical AI problems, the feasibility of using conceptual spaces has been studied in various application domains of AI, such as geographical measurement (Schwering & Raubal, 2005; Adams & Raubal, 2009a), cognitive robotics (LeBlanc, 2010; Cubek, Ertel, & Palm, 2015; Chella, Frixione, & Gaglio, 2003), object recognition (Brezolin, Fiorini, de Borba Campos, & Bordini, 2015), and visual perception (Chella, Frixione, & Gaglio, 1997). A recent review (Zenker & Gärdenfors, 2015) discusses further applications in diverse research areas (semantic spaces, computing meanings, and philosophical perspectives).

Concept formation tightly connects the theory of conceptual spaces to the induction (and particularly *learning*) problem. The aim of many *learning* systems is a general description of a category of observations as *concepts* (Luger, 2005). If the input of a learning algorithm takes the form of instances, attributes, and concepts, then the process of learning is called *concept description*. Instance-based learning refers to a class of learning algorithms which predicate the labels for the unseen instances based on their similarity to the nearest training instances (Keogh, 2011). This model requires a similarity function to perform the task of concept descriptions. However, in instance-based learning, the similarity functions are usually applied within a single-domain full feature space (Aha, Kibler, & Albert, 1991).

A comparison of practicality and effectiveness in instance-based learning and conceptual spaces was presented by Lee and Portier (2007). Yet the authors did not include the model construction process in their discussion. In this paper, we propose an instance-based approach for concept formation that considers the role of the involved features to derive a multi-domain space and represent the concepts in such a space.

Involving data mining approaches in the process of deriving conceptual spaces has been seldom studied. Keßler (2006) outlined the idea of using conceptual spaces to describe data, with some discussions on the possibility of automatically generating such spaces from databases. Lee (2005) proposed a data mining method coupled with conceptual space, which addresses cognitive tasks such as concept formation using, e.g., clustering techniques. The main drawback of these approaches is that, in order to directly set up the domains, they rely on knowing about the semantics of the field beforehand. However, an essential challenge is to automatically provide interpretable features as integrated quality dimensions. This work attempts to identify a meaningful set of features out of predefined features in the data set, relying on the hypothesis that most discriminative features of concepts (classes) are the most representative quality dimensions in the conceptual space.

An open question in the field of conceptual spaces is how quality dimensions are identified (Gärdenfors, 2000). Once the process of constructing a conceptual space starts, as Quine (1969) noted, some innate quality dimensions are needed to make *concept learning* possible. However, there is no standard way to specify which set of initiated dimensions are cognitively sufficient to characterise the concepts to be learned. In many developed examples of conceptual spaces, determining the set of quality dimensions relies on the background knowledge, which comes from human perceptual or sensory dimensions. This question is more challenging when there is no prior knowledge to explain the semantics of dimensions, or there is a lack of knowledge about relevant quality dimensions (Gärdenfors, 2004). This specific challenge motivates our investigation on how to derive the domains and quality dimensions in a data-driven manner.

Enabling data-driven domain creation for conceptual spaces extends their usefulness to problems they are not typically used for. However, keeping in mind the issues of induction and semantic inference in AI, the proposed approach particularly has the ability to deal with a class of learning problems that need a transparent interpretation for the overall model (not only interpretability of the decisions made).

2.3 Related Work on Semantic Inferences and Linguistic Descriptions

Revealing a semantic representation of unknown information connects *linguistic description* approaches to the present study, in a sense that one solution to determine content for data-driven information is to semantically model descriptive features of non-linguistic information and infer meaningful descriptions from the proposed model (Matthiessen, 1990).

Linguistic descriptions are the results of a natural language characterisation that covers the understandable information derived from the input data. This process is dependent on the semantic level of the informative features (attributes) that characterise and explain the data. This information can be obtained from various sources such as observations, sensor measurements, mathematical analysis or visual perceptions (Batyrshin, Sheremetov, & Herrera-Avelar, 2007). Importing *descriptive features* to computational systems includes the possibility of operating with linguistic information (Batyrshin et al., 2007).

The field of *linguistic description of data* (LDD) has emerged from the use of *fuzzy set theory* and soft computing to perform linguistic computations on data, which studies the methods for automatically describing numeric data sets by employing a set of linguistic terms (Ramos-Soto, Bugarín, & Barro, 2016). Fuzzy set theory is a well-studied approach to bridge between numeric and linguistic information, specifically in perception-based systems (Batyrshin et al., 2007; Kacprzyk & Zadrożny, 2010). The basic idea of linguistic descriptions comes from the works of Zadeh (2001) and Yager (1982) on *computing with words*, and later, *computational theory of perception* paradigms (Zadeh, 2000, 1996) to express the ability of computing systems in a linguistic manner (Diaz-Hermida, Pereira-Fariña, Vidal, & Ramos-Soto, 2016; Trivino & Sugeno, 2013). Within these paradigms, the promising tools are based on fuzzy quantification models to generate simple linguistic summaries of variables, such as "*most of the temperatures are hot*" (Delgado, Ruiz, Sánchez, & Vila, 2014). The fuzzy granulation provides propositions like *low*, *increasing*, *significant*, *near future* for the numerical features (Zadeh, 1997).

The studies on linguistic description also encompass the field of *natural language generation* (NLG). An NLG system generates human-readable natural language (Reiter, Dale, & Feng, 2000) from non-linguistic information. *Content determination* is the most relevant aspect of NLG for the linguistic characterisation of numeric data (Yu, Reiter, Hunter, & Mellish, 2007; Reiter, Sripada, & Robertson, 2003), which is the task of deriving interpretable information from the data to allow the system to reason symbolically instead of numerically (Reiter, 2007). Determining suitable content is mostly addressed using knowledge-driven approaches in order to comply with the domain or user requirements (Yu et al., 2007). Furthermore, Gkatzia (2015) reports that rule-based methods are dominant for current content selection approaches. However, when the content determination is performed in a data-driven manner, as in this work, it is possible to reveal unknown information, which can be beyond the user requirements, though still meaningful to represent (Banaee & Loutfi, 2014).

Since conceptual spaces are developed to model the attributes of concepts for further reasoning, it can be employed as a robust framework to perform content determination using semantic inferences. The problem of modelling natural language using conceptual spaces is seldom investigated in the literature (Agostaro et al., 2005; Derrac & Schockaert, 2015). Aisbett, Rickard and Gibbon (2015) recently investigated the integration of conceptual spaces theory with the topic of computing with words by introducing a fuzzy representation of conceptual spaces' elements. Domains and dimensions in their work, however, are crisp elements with no role concerning the qualification of objects within the space. Also, Derrac and Schockaert (2015) attempted to derive the semantic relations within conceptual spaces built upon text documents. However, to the best of our knowledge, there is no study on the conceptual spaces to derive natural language descriptions for the numeric inputs through a conceptual representation. The advantage of using conceptual spaces over the fuzzy set theory (as the primary approach for linguistic description) is in the content determination task, where the inference of linguistic descriptions for unknown observations can be easily modelled. This description will include both similar concepts and representative features from a multi-domain space, while in the learned fuzzy rules, the linguistic description of observations includes the fuzzy linguistic labels (Aisbett et al., 2015).

3. Data-Driven Construction of Conceptual Spaces

This section presents how to automatically construct a conceptual space from a numeric data set. Our approach to constructing conceptual spaces is called data-driven because it is automatically constructed by processing the data matrix of the observations based on the variable values and class labels. This way of construction is in contrast with the knowledge-driven conceptual spaces that have to be manually constructed using psychologically or scientifically pre-defined knowledge about the relations between quality dimensions, domains and the concepts' regions (Gärdenfors, 2004; Agostaro et al., 2005).

The process of constructing a conceptual space is about determining its essential elements. According to Aisbett and Gibbon (2001), and Raubal (2004), the definition of a conceptual space is as follows:

Definition 1. A *conceptual space* S is defined as a 4-tuple $\langle \mathcal{Q}, \Delta, \mathcal{C}, \Gamma \rangle$, where $\mathcal{Q}$ is a set of *quality dimensions*, Δ is a set of *domains*, $\mathcal{C}$ is a set of *concepts* in the space S, and Γ is a set of *instances* representing the concepts.

The representations of the elements are rigorously explained in further definitions (from 2 to 5). In order to automate the process of constructing conceptual spaces, the definitions of the conceptual spaces' elements are slightly modified compared to previous formulations (Adams & Raubal, 2009b; Rickard et al., 2007). These modifications consider the fact that the constructed conceptual space will be utilised as a semantic representation model for further inferences.

To start the constructing process, we assume that a given data set $\mathcal{M}$ contains a set of possible class labels, a set of predefined features, and the input observations with known class labels, which are characterised by feature values. Formally, given a set of class labels $\mathcal{Y} = \{y_1, \ldots, y_m\}$ and a set of features $\mathcal{F} = \{X_1, \ldots, X_n\}$, let $\mathcal{D}$ be the set of known observations, denoted by $\mathcal{D} = \{o_i : (\mathbf{x}_{o_i}, y_{o_i})\}$, where o_i consists of a n-dimensional feature vector $\mathbf{x}_{o_i} = [x_1, \ldots, x_n]$, and an output label $y_{o_i} \in \mathcal{Y}$. The component x_j $(j = 1, \ldots n)$ in the vector $\mathbf{x}_{o_i}$ is the measured value of the corresponding feature $X_j \in \mathcal{F}$.

Here, each feature X is defined as a couple of values $X : \langle H_X, I_X \rangle$, where H_X indicates the linguistic name of the feature, and I_X is either a numeric interval or a categorical set that presents the possible range of values for X.

Example 1. Consider the *leaf data set* (Silva, Marcal, & da Silva, 2013) which is a set of photographed leaf samples (observation set $\mathcal{D}^l$) from various plant species (classes) such as: $\mathcal{Y} = \{$ $y_{\texttt{qr}}$: '`Quercus Robur`', $y_{\texttt{ap}}$: '`Acer Palmatum`', $y_{\texttt{no}}$: '`Nerium Oleander`', $y_{\texttt{tt}}$: '`Tilia Tomentosa`', …$\}$. This data set includes a set of measurable features to characterise the features of each leaf sample, such as: $\mathcal{F} = \{$ $X_{\texttt{el}}$: `elongation`, $X_{\texttt{lo}}$: `lobedness`, $X_{\texttt{co}}$: `convexity`, $X_{\texttt{ro}}$: `roundness`, $X_{\texttt{so}}$: `solidity`, $X_{\texttt{in}}$: `indentation`, …$\}$. So, an observed leaf such as $o_i \in \mathcal{D}^l$ that is labelled by $y_{\texttt{tt}}$ takes the feature values as: $o_i : (\mathbf{x}_{o_i}, y_{\texttt{tt}})$, where $\mathbf{x}_{o_i} = [x_{\texttt{el}}, x_{\texttt{lo}}, x_{\texttt{co}}, x_{\texttt{ro}}, x_{\texttt{so}},\ x_{\texttt{in}}]$.

The goal of this section is to find a mapping from the elements of a data set $\mathcal{M}$ to various elements needed to define a conceptual space S. In short, this mapping is achieved by performing the following steps:

- Initialise the primitive known concepts using the class labels. Consequently, the conceptual space S, which models the data set $\mathcal{M}$, will consist of a set of concepts $\mathcal{C} = \{C_1, \ldots, C_m\}$, where $|\mathcal{C}| = |\mathcal{Y}|$. Thus, the notation C_y indicates the concept which corresponds to the class label $y \in \mathcal{Y}$.[1]
- Specify the quality dimensions $\mathcal{Q}$ and domains Δ. The quality dimensions are specified via selecting a subset of the features such that $\mathcal{Q} \subset \mathcal{F}$, and the domains are determined based on ranking and grouping the set of selected features as the quality dimensions (Section 3.1).
- Form the representation of each concept C_y within the derived domains Δ, based on the known corresponding instances (Section 3.2).

Figure 1 illustrates the steps of constructing a conceptual space from a set of numeric data, which are explained in the following sections.

3.1 Domain and Quality Dimension Specification: A Feature Selection Approach

A data-driven approach to build a conceptual space makes no prior assumption about the domains. Rather, the known labelled observations and features are the inputs from which the quality dimensions and domains will be extracted. This approach aims to propose a set of observation-based associations between classes of data as concepts and the grouped subsets of features as domains. As a domain is an integrated subset of quality dimensions,

[1] This approach follows that any semantic modelling needs an innate set of knowledge (Quine, 1969). Learning the concepts from scratch is not the scope of this work.

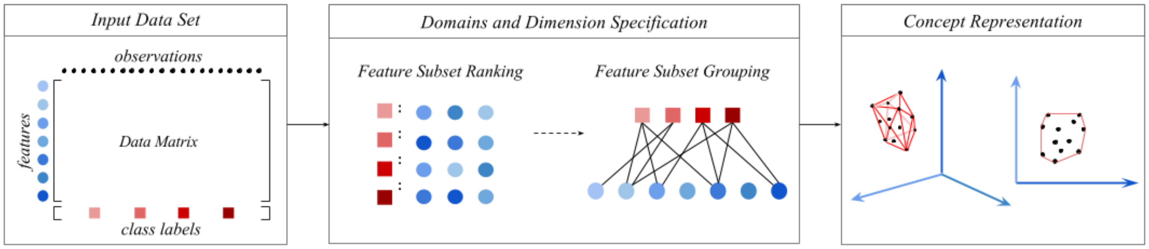


Figure 1: Illustration of the main steps for constructing a conceptual space from a set of numeric data. The domain and dimension specification is explained in Section 3.1, and the concept representation is described in Section 3.2.

and the quality dimensions are the subset of the initialised features, the first step is to determine a subset of informative features. This determination is performed by applying *feature selection* methods. Before explaining these methods, we recall the formal definitions of a quality dimension and a domain. As mentioned before, these definitions are reshaped in a novel manner to be utilised in the task of semantic inference.

Definition 2. A *quality dimension* $q_X \in \mathcal{Q}$ is a triple $\langle H_q, I_q, \mu_q \rangle$, which corresponds to a selected feature $X \in \mathcal{F}$. H_q is the linguistic name of the quality dimension q_X, which is equal to H_X and I_q indicates the range of possible values for the quality dimension q_X, which is equal to I_X. μ_q is defined as a family of fuzzy membership functions[2] to map the subintervals of I_q onto a set of linguistic terms.

Definition 3. A *domain* δ is a triple $\langle \mathcal{Q}(\delta), \mathcal{C}(\delta), \omega_\delta \rangle$, where $\mathcal{Q}(\delta) \subset \mathcal{Q}$ is the set of integral quality dimensions involved in δ, $\mathcal{C}(\delta) \subset \mathcal{C}$ is the set of concepts that are represented in δ, and $\omega(\delta)$ is a weight function[3] presenting the assigned salient weight between a concept and a quality dimension in δ.

Example 2. Consider the leaf data set from Example 1, suppose that a quality dimension is *elongation*, which is defined as $q_{\texttt{el}} = \langle$ ‘`elongation`’, $[0, 1]$, $\mu_{\texttt{el}} \rangle$, and another one is *lobedness*, defined as $q_{\texttt{lo}} = \langle$ ‘`lobedness`’, $(0, \inf)$, $\mu_{\texttt{lo}} \rangle$. One can conceptualise the leaves in various domains such as *Shape*, *Texture*, *Colour*, etc. Then, both *elongation* and *lobedness* quality dimensions can belong to the *shape* domain. Moreover, $\mu_{\texttt{el}}$ can return the linguistic labels for *elongation* as: ‘*circular*’, ‘*elliptical*’, ‘*elongated*’.

Since we construct the domains in a data-driven way without involving prior knowledge, it can be difficult to assign a semantic interpretation to the constructed domains, which reflects human perception. However, the provided space is counted as a conceptual space because of its ability to represent the concept formation and the semantic similarities between concepts and instances across the domains. Thus, within the proposed model, the set of quality dimensions of a domain is an integral subset of all quality dimensions which are analytically relevant or dependent to each other.

[2] More details on the fuzzy membership functions μ, which quantify the changes in the values of a feature by assigning linguistic labels to the subintervals of dimension, will be given in Section 4.2.1.

[3] The weight function $\omega(\delta)$ is further explained in Section 3.1.2.

Identifying the most characteristic features of the data from the initialised set of features is an essential task, which is generally performed by *feature extraction* approaches (Bengio, Courville, & Vincent, 2013). There are two principal ways to extract informative features: *feature transformation* and *feature selection* (Guyon, Gunn, Nikravesh, & Zadeh, 2008). The first approach finds a projection from the original feature space into a lower dimensional feature space. Transforming the original features into this lower dimensional space usually alters any associated descriptive attributes connected to the features. Therefore, the semantic meaning of the resulting features is often difficult, if not impossible, to assess (Guyon & Elisseeff, 2003). The second approach selects a subset of original features by keeping relevant features and discarding the irrelevant ones. The retained features are not altered, and the original semantic meaning of those features stays intact. Since our goal is to exploit external knowledge of the original features, we apply *feature subset selection* techniques.

Both *relevance* and *redundancy* are essential criteria to consider in feature selection. A subset of features is optimal if the relevance between selected features and the target classes is maximal, and the redundancy among the selected features is minimal. These two criteria guarantee that the selected features are adequate to distinguish the classes of data with the smallest number of features (Duch, 2006).

The proposed approach employs feature selection methods to specify the quality dimensions and domains in two phases: *feature subset ranking* and *feature subset grouping.* The feature subset ranking phase determines which features are most representative for every single target class, independent from other classes. Since a concept can rather be represented by one or several groups of features as domains, the feature subset grouping phase categorises the ranked features in a way to recognise what features are most related to each other based on their relevancy to the concepts. The following subsections show the two phases of the proposed solution, (1) feature subset ranking and (2) feature subset grouping, in order to specify the suitable quality dimensions within a set of domains.

3.1.1 Feature Subset Ranking

Feature subset selection algorithms are categorised as either *filter* methods or *wrapper* methods (Witten & Frank, 2005). *Filter* methods determine the subset of features based on the statistical characteristics of the input data set without referring to the used classifier. *Wrapper* methods are dependent on the learning algorithm (i.e., target classifier) that evaluates the selected subset of features based on the performance of the used learning algorithm. In the present work, the aim is to identify the meaningful set of understandable attributes out of predefined features, but not to classify the input data. So, *filter* methods are chosen to be used for feature selection, since this category of methods is independent of the final classifier approach and it derives an informative subset of features with respect to the input data set labels.

Filter methods rank the features using a scoring function, usually by employing a statistical or information theoretical measure to quantify relevance and redundancy. The top scored features are kept as selected features (or low scored ones are removed from the resulting subset). In this work, *mutual information*, one of the commonly employed scoring functions, is used (Brown, 2009). One such technique is *MIFS* (mutual information-based feature selection) (Battiti, 1994). For an input set of features $\mathcal{F}$ and 2-class labelled data $\mathcal{D}$, MIFS adds the feature $X_i \in \mathcal{F}$ to the already chosen subset of features $\mathcal{F}'$, in order to maximise

$$I(\mathcal{D}, X_i) - \beta \sum_{X_j \in \mathcal{F}'} I(X_i, X_j), \tag{1}$$

where $I(Y, X)$ is the mutual information[4] between the variables Y and X (Torkkola, 2008). The first term in equation 1 attempts to maximise the relevance of feature X_i to target labelled data, and the second term tries to minimise the redundancy between X_i and the already selected features in $\mathcal{F}'$ (using a balancing parameter β). In our work, the term $I(Y, X)$ is estimated via histograms, but other estimation methods are applicable as well (Schaffernicht, Kaltenhaeuser, Verma, & Gross, 2010). The MIFS technique is generally a heuristic approximation since there is no independent assessment of the *joint mutual information* to determine when a feature is relevant to the class labels (Torkkola, 2008). The proposed method is not dependent on the use of the MIFS algorithm. It can be substituted by other approximations of the joint mutual information (Fleuret, 2004; Peng, Long, & Ding, 2005; Brown, Pocock, Zhao, & Luján, 2012). It is notable that different filter methods do not necessarily produce the same ranking of the features. However, the focus here is to reach to a good enough set of features representing the data classes with high relevance and low redundancy (Webb, 2003).

The proposed method for feature subset ranking starts with defining a new set of input data for each target label y, wherein the data set of known observations $\mathcal{D}$ is split into two classes: class y including all the observations labelled by class y, denoted by $\mathcal{D}_y$, and class $\overline{y} = \{\mathcal{Y} \backslash y\}$ including the rest of observations labelled by other classes than class y, denoted by $\mathcal{D}_{\overline{y}}$. Then the MIFS procedure is applied to the feature set $\mathcal{F}$ considering the generated 2-class data set $\mathcal{D}_{y\overline{y}} = \{\mathcal{D}_y \cup \mathcal{D}_{\overline{y}}\}$. By separating one class (concept) of data from the other classes, the output of the feature ranking algorithm will return the features that individually characterise the observations of this concept and separate it from the rest. The output of the filter method for each label is a sorted list of features with a score for each feature. Formally, the output for a class y is a ranked list of features with the highest scores according to y, as

$$\mathcal{R}(y) = \{\ (X, w_{y,X}) \mid X \in \mathcal{F}\ ,\ w_{y,X} \in [0,1]\ \} \tag{2}$$

where $w_{y,X}$ is the normalised weight (or the score) that is assigned to the relation of feature X and label y. The features in $\mathcal{R}(y)$ are the k most relevant features of the class label y. From a conceptual point of view, these k features of $\mathcal{R}(y)$ are the suitable candidates to be the quality dimensions that distinguish the concept C_y from the other concepts. The score $w_{y,X}$ determines the importance of the selected feature X to represent the class label y. From the conceptual space point of view, the scores indicating the weights show the significance of the chosen quality dimensions for C_y.

Algorithm 1 shows the steps for finding the ranked scored list of features for each label y. The output of the algorithm is then a set of filter method results for all the class labels, denoted by $\mathcal{R} = \{\mathcal{R}(y_1), \ldots, \mathcal{R}(y_m)\}$. In this algorithm, $\mathcal{F}'$ is the set of all features that

[4] This mutual information is defined based on the probability density functions, denoted: $I(X, Y) = \int_x \int_y p(x, y) \log \frac{p(x,y)}{p(x)p(y)} dydx$ (Cover & Thomas, 2006).

Algorithm 1: Feature Subset Ranking

```
Function FeatureRanking($\mathcal{D}, \mathcal{Y}, \mathcal{F}$)
    $\mathcal{R}, \mathcal{F}' \leftarrow \emptyset$
    foreach $y \in \mathcal{Y}$ do
        // Define 2-class data set
        $\mathcal{D}_y = \{o_i : (\mathbf{x}_i, y_i) \in \mathcal{D} \mid y_i = y\}$
        $\mathcal{D}_{\overline{y}} = \{o_i : (\mathbf{x}_i, y_i) \in \mathcal{D} \mid y_i \neq y\}$
        $\mathcal{D}_{y\overline{y}} \leftarrow \mathcal{D}_y \cup \mathcal{D}_{\overline{y}}$
        // Find the list of top scored features $(X, w_{y,X})$
        $\mathcal{R}(y) \leftarrow MIFS(D_{y\overline{y}}, \mathcal{F})$
        $\mathcal{F}' \leftarrow \mathcal{F}' \cup \{X \mid \exists\ (X, w_{y,X}) \in \mathcal{R}(y) \wedge X \in \mathcal{F}, w \in [0,1]\}$
    return $\mathcal{R}$, $\mathcal{F}'$
```

appeared (at least once) in the ranked features:

$$\mathcal{F}' = \bigcup_{y \in Y} \{X \mid \exists\ (X, w_{y,X}) \in \mathcal{R}(y) \wedge X \in \mathcal{F}, w_{y,X} \in [0,1]\}, \tag{3}$$

where $\mathcal{F}' \subset \mathcal{F}$. The set of features $\mathcal{F}'$ is the set of features to become quality dimensions. However, in feature grouping, some of these features may be filtered out from the target set of quality dimensions. Note that the time complexity of the feature subset ranking (MIFS) is quadratic (Bollacker & Ghosh, 1996) depending on the number of initial features.

Example 3. Continuing of the leaf data set in Example 1, suppose that after applying the MIFS method, *elongation*, *lobedness*, and *roundness* are selected as the top features for $y_{\texttt{tt}}$, $\mathcal{R}(y_{\texttt{tt}}) = \{\ (X_{\texttt{el}}, w_{y_{\texttt{tt}},X_{\texttt{el}}}), (X_{\texttt{lo}}, w_{y_{\texttt{tt}},X_{\texttt{lo}}}), (X_{\texttt{ro}}, w_{y_{\texttt{tt}},X_{\texttt{ro}}})\}$. Also, *elongation*, *roundness*, and *indentation* are selected for $y_{\texttt{no}}$, $\mathcal{R}(y_{\texttt{no}}) = \{\ (X_{\texttt{el}}, w_{y_{\texttt{no}},X_{\texttt{el}}}), (X_{\texttt{ro}}, w_{y_{\texttt{no}},X_{\texttt{ro}}}), (X_{\texttt{in}}, w_{y_{\texttt{no}},X_{\texttt{in}}})\}$. Then, $\mathcal{F}' = \{X_{\texttt{el}}, X_{\texttt{lo}}, X_{\texttt{ro}}, X_{\texttt{in}}\}$.

3.1.2 Feature Subset Grouping

Based on the definitions in conceptual space theory, a quality dimension usually appears in a single domain along other relevant dimensions to represent a specific aspect of conceptualised observations (Gärdenfors, 2000; Zenker & Gärdenfors, 2015). It might be possible to have the same dimension in various domains, but this requires a priori knowledge (Banaee & Loutfi, 2014). Moreover, repeating dimensions in a multi-domain space increases the redundancy, and consequently decreases the distinction of the domains for a meaningful concept representation. Therefore, the selected features are divided into distinct partitions of features as target domains, in order to avoid either creating a single domain of full features, or repeating features in all of the constructed domains.

Here we propose a heuristic method to detect distinct subsets of features, where the features in each subset are highly representative of the most relevant classes. The output set $\mathcal{R}$ in Algorithm 1 is a set of ranked features for each label. It is obvious that some features might be repeated in the ranked set of different class labels in $\mathcal{R}$. From the information in set $\mathcal{R}$, the goal is to extract those subsets of features that are associated to each other based on their co-appearance in the ranked features of each class. We first introduce a

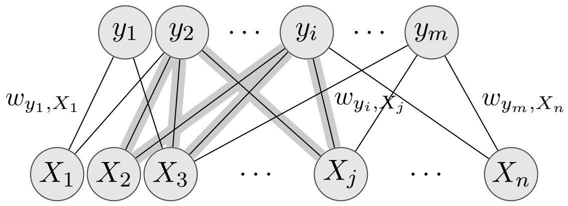


Figure 2: A weighted bipartite graph with two sets of vertices from the labels $\mathcal{Y}$ and the selected features $\mathcal{F}'$. Also, an example of a biclique shown in the highlighted edges.

graph representation of the label-feature relation, and then derive the correlated features using a greedy search on this graph. More specifically, we build up a bipartite graph and search for the bicliques that identify the most associated feature subsets (i.e., domains).

Let $G = (V_\mathcal{Y} \cup V_{\mathcal{F}'}, E, w)$ be a bipartite graph with two sets of vertices V_Y and $V_{\mathcal{F}'}$, a set of edges E, and $w : V_\mathcal{Y} \times V_{\mathcal{F}'} \to \mathbb{R}$ as a weight function for the edges. The vertex set $V_\mathcal{Y}$ denotes the class labels in $\mathcal{Y}$. The vertex set $V_{\mathcal{F}'}$ denotes the top-ranked features in $\mathcal{F}'$. A vertex $v_y \in V_\mathcal{Y}$ is connected to a vertex $v_X \in V_{\mathcal{F}'}$ if $X \in \mathcal{F}'$ has been selected for $y \in \mathcal{Y}$ in Algorithm 1. In other words, for each pair $(X, w_{y,X}) \in \mathcal{R}(y)$ a new edge $v_y v_X$ is added to the edge set E of bipartite graph G between vertices v_y and v_X, where the weight of the edge $v_y v_X$ is denoted by $w(v_y v_X) = w_{y,X}$. Figure 2 is an illustration of such weighted bipartite graph G.

The idea of grouping features is to find the maximal connected subgraphs in G. More precisely, a subset of features which are all connected to the same set of classes is a suitable subset of features for feature grouping. A biclique $\hat{G} \subset G$ is a special bipartite graph where every vertex in one part of vertices is connected to all the vertices in the other part of the vertices. The highlighted edges in Figure 2 depicts an example of a biclique in the given bipartite graph. Let $\hat{G}$ be a biclique denoted by $\hat{G} = (\hat{V}_\mathcal{Y} \cup \hat{V}_{\mathcal{F}'}, \hat{E}, w)$, where $\hat{V}_\mathcal{Y} \subset V_\mathcal{Y}$, $\hat{V}_{\mathcal{F}'} \subset V_{\mathcal{F}'}$. In this biclique, assume $|\hat{V}_\mathcal{Y}| = \hat{m}$, $|\hat{V}_{\mathcal{F}'}| = \hat{n}$, thus $|\hat{E}| = \hat{m} \times \hat{n}$. The proposed approach is looking for a biclique with the highest score as $\hat{G}_{max}$ among all the bicliques in G. The score of a biclique $\hat{G}$ is calculated using a scoring function $Score_{\hat{G}}$, based on the weights of its edges, as follows:

$$Score_{\hat{G}} = \sum_{v_y \in \hat{V}_\mathcal{Y}} \Big(\prod_{v_X \in \hat{V}_{\mathcal{F}'}} w(v_y v_X) \Big) \; / \; \hat{n} \tag{4}$$

This scoring function calculates the average of the weights associated in the biclique. Generally, this scoring function will return higher values for the bicliques with a higher number of class labels and lower number of features.

In the selected biclique ($\hat{G}_{max}$), the involved features $\hat{V}_{\mathcal{F}'}$ then will be the subset of features as the quality dimensions of a domain δ. To identify the next domain, the set of features $\hat{V}_{\mathcal{F}'}$ is eliminated from the graph G, since these features are already assigned to a domain. After that, the process of finding the best biclique repeats on the updated graph G to find the next maximal biclique. Algorithm 2 shows the steps of determining the domains with feature subset grouping. In general, the problem of finding maximum edge biclique in a bipartite graph is an NP-complete problem (Shaham, Yu, & Li, 2016). However, the

Algorithm 2: Feature Subset Grouping

```
Function FeatureGrouping($\mathcal{R}$, $\mathcal{F}'$, $\mathcal{Y}$)
    $\Delta, Q \leftarrow \emptyset$
    // Build bipartite graph
    $V_{\mathcal{Y}} \leftarrow Y$
    $V_{\mathcal{F}'} \leftarrow \mathcal{F}'$
    foreach $(X, w_{y,X}) \in \mathcal{R}(y) : y \in \mathcal{Y}$ do
        $E \leftarrow E \cup v_y v_X$
        $w(v_y v_X) \leftarrow w_{y,X}$
    $G = (V_{\mathcal{Y}} \cup V_{\mathcal{F}'}, E, w)$
    // Find max cliques as domains
    do
        $\hat{G}_{max}(\hat{V}_{\mathcal{Y}} \cup \hat{V}_{\mathcal{F}'}, \hat{E}, w) \leftarrow MaxBiclique(G)$
        if $\hat{G}_{max} = \emptyset$ then
            break
        $\delta : \langle\ \mathcal{Q}(\delta), \mathcal{C}(\delta), \omega_\delta\ \rangle \leftarrow \langle \hat{V}_{\mathcal{F}'}, \hat{V}_{\mathcal{Y}}, w \rangle$
        $\Delta \leftarrow \Delta \cup \delta$
        $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{Q}(\delta)$
        // Update bipartite graph
        $G \leftarrow \big(V_{\mathcal{Y}} \cup (V_{\mathcal{F}'} \backslash \hat{V}_{\mathcal{F}'}), (E \backslash \hat{E}), w\big)$
    until $(\mathcal{Q} = \mathcal{F}')$
return $\Delta$, $\mathcal{Q}$
```

developed approximation method for feature grouping approach has polynomial time complexity (depending on the number of classes, not features) due to the fact that the degree of the class vertices in the constructed bipartite graph is a constant number (k). Thus, by scaling the features size or the class size, the approach is still computationally feasible.

As stated in Definition 3, a domain δ is a triple $\langle\ Q(\delta), C(\delta), \omega_\delta\ \rangle$. The weight function $\omega_\delta = \mathcal{C}(\delta) \times \mathcal{Q}(\delta) \rightarrow \mathbb{R}$ is a function presenting the assigned salient weight between a concept $C_y \in \mathcal{Y}(\delta)$ and a quality dimension $q_X \in \mathcal{Q}(\delta)$. For a chosen biclique $\hat{G}_{max} = (\hat{V}_{\mathcal{Y}} \cup \hat{V}_{\mathcal{F}'}, \hat{E}, w)$, we can construct a domain δ as a triple $\langle\ \mathcal{Q}(\delta), \mathcal{C}(\delta), \omega_\delta\ \rangle = \langle\ \hat{V}_{\mathcal{F}'}, \hat{V}_{\mathcal{Y}}, w\ \rangle$. More specifically, for a constructed domain $\delta = \langle\ \mathcal{Q}(\delta), \mathcal{C}(\delta), \omega_\delta\ \rangle$, the set of quality dimensions is $Q(\delta) = \{q_X \mid v_X \in \hat{V}_{\mathcal{F}'}\}$. Also, the set of concepts related to δ is defined as $C(\delta) = \{C_y \mid v_y \in \hat{V}_{\mathcal{Y}}\}$. Then $\omega_\delta(C_y, q_X) = w(v_y v_X)$.

It is worth noting that for the next iteration of finding bicliques, the vertices with the labels $\hat{V}_{\mathcal{Y}}$ of a chosen biclique are not eliminated while updating the bipartite graph, because a class label can be involved in other bicliques in further iterations. From a conceptual space point of view, it is also meaningful, since a concept can be represented in several domains.

Example 4. The corresponding bipartite graph to the ranked features from Example 3 is illustrated in Figure 3. In this bipartite graph, one biclique is highlighted, which can potentially be the best biclique. If so, then the features *elongation* and *roundness* will become the quality dimensions of a new domain δ as: $\mathcal{Q}(\delta) = \{q_{\texttt{el}},\ q_{\texttt{ro}}\}$ and $\mathcal{C}(\delta) = \{C_{\texttt{tt}}, C_{\texttt{no}}\}$.

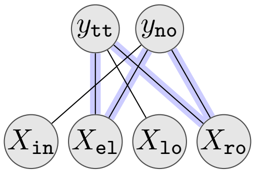


Figure 3: A bigraph graph and one selected biclique (blue edges) for the leaf example (explained in Example 4).

3.2 Concept Representation: An Instance-Based Approach

The essential concern to represent a concept in a conceptual space is to decide which are the most relevant quality dimensions and consequently most relevant domains. A concept may be represented in one domain or in several domains. The important point is that a concept is not necessarily associated with a certain subset of domains, but usually, one domain or a few numbers of domains are prominent to represent a concept (Gärdenfors, 2000). In Section 3.1, the selected features are grouped into a set of domains out of the extracted bicliques. Using the fact that each class label will be involved in at least one selected biclique, then the corresponding concept is assigned to (or associated with) a certain number of domains (at least one). With the output of Algorithm 2 we already know which concepts are associated with which domains.

It is possible that one concept appears in two bicliques, which means the concept is relevant to both specified domains. This fact is consistent with the conceptual spaces theory since a concept is not always represented within a single domain. The typical example is the concept of '*apple*' which is represented with more than a single domain, such as *colour*, *taste*, *size*, etc. A concept with merely one related domain is called *property* (Gärdenfors, 2000). In fact, a property is a special form of a concept defined in a single domain (Gärdenfors & Warglien, 2012). For example, the colour '*green*' is a property which is represented only in the *colour* domain. Thus, a concept can be specified as a single property within a single domain (e.g., green), or as a collection of properties within several domains (e.g., apple). Depending on the domain specification process, a class label in the input data set might be represented either as a property in one domain or as a concept in several domains. The problem of deriving the domains in a data-driven manner is that for a concept represented in several domains, there is no trivial interpretation for the meaning of its properties within the domains. This issue comes from the fact that the interpretation of the data-driven domains themselves is also tricky.

For a set of concepts $\mathcal{C} = \{C_{y_1}, \ldots, C_{y_n}\}$, the problem is how to formulate the geometrical representation of concepts in the conceptual space with the extracted set of domains Δ. In general, a natural concept is a collection of regions across one or more domains along with a set of salient weights to the domains (Gärdenfors, 2000). For a concept C_y, let $\Delta(y) \in \Delta$ be a subset of domains that contain concept C_y in their concept sets, as $\Delta(y) = \{\delta_i | \ \delta_i \in \Delta \wedge C_y \in C(\delta_i)\}$, assuming that $|\Delta(y)| = k$. The concept C_y is presented by a collection of *sub-concepts*, denoted: $C_y = \{c_y^1, \ldots, c_y^k\}$, where each c_y^i is the representation of C_y within the domain $\delta_i \in \Delta(y)$.

Algorithm 3: Concept Representation

```
Input: $\mathcal{D}_y \subset \mathcal{D}$: set of observations that are labelled by $y \in \mathcal{Y}$ and $\Delta(y) \subseteq \Delta$: domains that
        contain $C_y$ in their concept sets.
Output: A Concept $C_y = \{c_y^1, \dots, c_y^k\}$, representing label $y$ in the conceptual space.
foreach $o \in \mathcal{D}_y$ do
    $\gamma_o \leftarrow Vectorise(o, \Delta(y), \mathcal{Q})$                // determine $p_\gamma^1, \dots, p_\gamma^k$ in $\delta_1, ..., \delta_k$
    $\Gamma(y) \leftarrow \Gamma(y) \cup \gamma_o$
foreach $\delta_i \in \Delta(y)$                                       // $\delta_i = \langle Q(\delta_i), C(\delta_i), \omega_{\delta_i} \rangle$ , $1 \leq i \leq k$
do
    $c_y^i \leftarrow \emptyset$
    foreach $\gamma \in \Gamma(y)$ do
        $P_y^i \leftarrow P_y^i \cup \{p_\gamma^i\}$
    $\eta \leftarrow ConvexHull(P_y^i)$
    $\phi \leftarrow \{\omega_{\delta_i}(C_y, q^i) | C_y \in \mathcal{C}(\delta_i), q^i \in \mathcal{Q}(\delta_i)\}$
    $c_y^i \leftarrow \langle \eta, \phi \rangle$
    $C_y \leftarrow C_y \cup c_y^i$
```

Definition 4. A *sub-concept* c_y^i, representing the concept C_y in the domain δ_i, is defined as a tuple $\langle \eta, \phi \rangle$, where η is the region representing the geometrical area of C_y in the domain δ_i, and ϕ is a set of weights indicating the assigned degrees of salience between C_y and each quality dimension $q \in \mathcal{Q}(\delta_i)$.

In order to represent a concept, we define the representation of its sub-concepts. The following two subsections describe the way to formally represent a concept, by defining its regions and its set of weights, respectively. Algorithm 3 shows the steps of the concept representation, with the required parameters to represent a concept C_y.

3.2.1 Convex Regions of Concepts

The identification of the geometrical regions of concepts is based on the location of the known observations. The concept $C_y \in \mathcal{C}$ is represented using the subset of observations $\mathcal{D}_y = \{o_1, o_2, \dots, o_{n_y}\}$ which are labelled with $y \in \mathcal{Y}$. We define $\Gamma(y)$ as the set of *instances* related to the observations in $\mathcal{D}_y$, denoted by $\Gamma(y) = \{\gamma_1, \gamma_2, \dots, \gamma_{n_y}\}$. These instances then specify the geometrical representation of the concept C_y as a set of regions within the domains. The set of all instances Γ in a conceptual space S is defined as: $\Gamma = \bigcup_{y \in \mathcal{Y}} \Gamma(y)$.

Definition 5. An *instance* γ related to the concept C_y is a finite set of n-dimensional points $\gamma = \{p_\gamma^1, \dots, p_\gamma^k\}$ with a one-to-one mapping from the instance points to the domains $\Delta(y)$, where $|\Delta(y)| = |C_y| = k$.

An instance $\gamma_o \in \Gamma(y)$ is the representation of the observation $o \in \mathcal{D}_y$. The points of γ_o are the values of the associated quality dimensions, which are stored in the feature vector $\mathbf{x}_o$. Formally, each point $p_\gamma^i \in \gamma_o$ in a domain $\delta_i \in \Delta(y)$ is a numeric vector of the values of the quality dimensions in δ_i, denoted: $p_\gamma^i =< q_1(\gamma_o), \dots, q_{|Q(\delta_i)|}(\gamma_o) >$, which is a sub-vector of the feature vector $\mathbf{x}_o$ that includes the features associated with the quality dimensions in $\mathcal{Q}(\delta_i)$. This process of determining the points of an instance γ_o using the feature vector of the observation o is called *vectorisation*.

Since all the instances with the label y have a point p^i in domain $\delta_i \in \Delta(y)$, to identify the convex region η of a sub-concept c^i_y, we need to know the location of all these points in the domain. Let P^i_y be the collection of all the points which their corresponding instances are labelled by y, and these points are located in domain δ_i. So, $P^i_y = \{p^i_{\gamma_1}, p^i_{\gamma_2}, \ldots, p^i_{\gamma_{n_y}}\}$, where $p^i_{\gamma_j} \in \gamma_j$, $\gamma_j \in \Gamma(y)$, and $j = 1...n_y$.

Example 5. Figure 4a consists of two domains δ_a and δ_b with two and three quality dimensions, respectively. Assume a concept C_y has two sub-concepts c^a_y and c^b_y within these domains. So, each instance $\gamma_j \in \Gamma(y)$ includes two points p^a_j and p^b_j located in the domains. Figure 4a depicts the set of points P^a_y and P^b_y, which will be used to represent sub-concepts c^a_y and c^b_y, respectively.

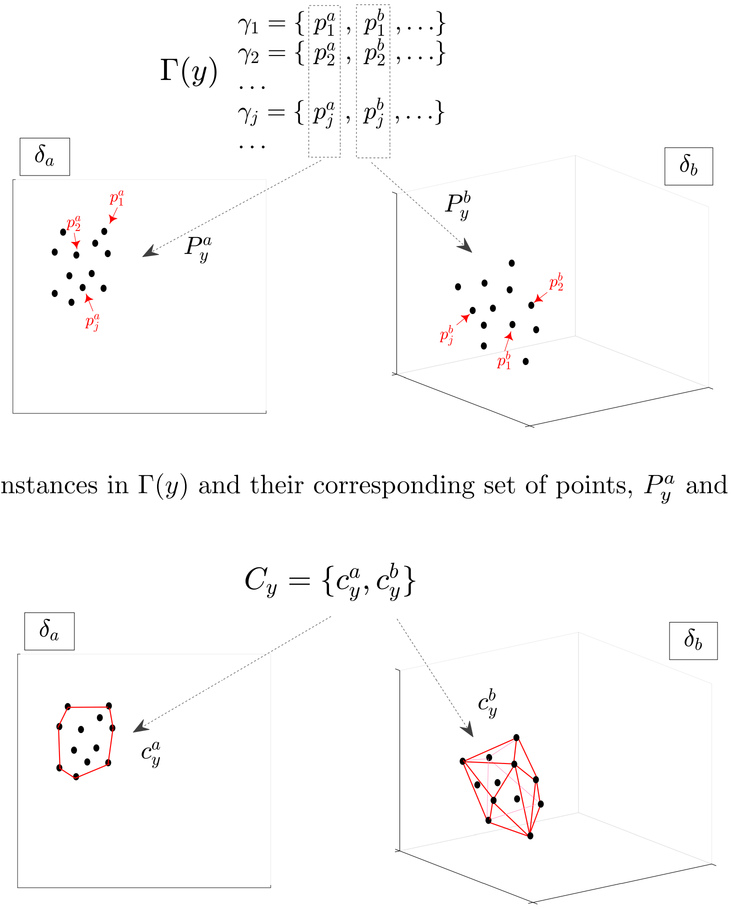
The *convexity*, *connectedness*, and *betweenness* are the geometrical criteria required to define a region for a concept in the theory of conceptual spaces (Gärdenfors, 2000). The convexity of concepts is crucial to facilitate the *learnability* of concepts through the instances (Gärdenfors, 2014). A convex region is a geometric structure within a multidimensional domain which satisfies convexity and connectedness criteria. There are various approaches to identify the convex region covering a set of giving points, such as *convex hull* and *Voronoi tessellations* algorithms, or defining an ellipsoid around the points (Gärdenfors, 2000; Adams & Raubal, 2009b). For our purpose, the convex hull (CH) is a more convenient choice among others because it also satisfies the *betweenness* criterion. Since the concepts' regions are formed by its instances, both ellipsoid and Voronoi regions assign some points of the space to a concept's region, which are not necessarily between the concept's known instances. For a sub-concept c^i_y, the convex region η is defined as the convex hull (i.e., convex polytope) of the points in P^i_y, as: $\eta(c^i_y) = CH(P^i_y)$.

Example 6. Figure 4b shows the convex regions of the sub-concepts c^a_y and c^b_y. The convex hull $\eta(c^a_y)$ is a 2D polygon in δ_a, and $\eta(c^b_y)$ is a 3D polytope in δ_b. These convex hulls are calculated based on the points P^a_y and P^b_y from the instances in $\Gamma(y)$ in Figure 4a.

3.2.2 Context-Dependent Weights of Concepts

Depending on the *context*, the *salience* given to various aspects of a concept may vary (Gärdenfors, 2000). In the example of the *apple* concept, in one context the *taste* domain might be more prominent, but in another context, *shape* domain can be salient. In contrast, in such examples of concepts that there is no common knowledge about the salience of the domains in various concepts, the data itself determines the salience of domains and quality dimensions and defines the context-based weights for the concepts. In other words, the observations from different contexts define which domain and quality dimensions are more important to represent the given concepts.

Example 7. For the example of *leaf* data set, suppose that *shape* and *colour* are the domains and suppose that we want to differ between the contexts of *Swedish* leaves and *Japanese* leaves. Knowing the common-sense knowledge about these contexts might be useless to determine the weights of the domains and dimensions. However, based on the observed data in each of these contexts, one can realise that e.g., the quality dimensions in



(a) Instances in $\Gamma(y)$ and their corresponding set of points, P_y^a and P_y^b.

(b) Convex regions of the sub-concepts c_y^a and c_y^b.

Figure 4: A concept representation example in a conceptual space with domains δ_a and δ_b.

the *shape* domain are more salient rather than the *colour* domain to represent the *Swedish* leaves, but this inference may not be necessarily valid for *Japanese* leaves.

These relative degrees of salience assigned to the dimensions of the domains implicitly represent the notion of context. Here, the context-dependent weights are already embedded in the representation by calculating the relevance of quality dimensions to the concepts (i.e., the weights of the bipartite graph) while specifying the domains. The salient weights ϕ for a sub-concept c_y^i come from the assigned weights ω_{δ_i} in δ_i between $C_y \in \mathcal{C}(\delta_i)$ and any quality dimension in $\mathcal{Q}(\delta_i)$. Formally:

$$\phi(c_y^i) = \{\omega_{\delta_i}(C_y, q^i) \mid C_y \in \mathcal{C}(\delta_i),\ q^i \in \mathcal{Q}(\delta_i)\}. \quad (5)$$

So, each sub-concept has its own set of weights in relation to the domain's quality dimensions. This point individualises the definition of context-dependent weights from the definition of context weights in other developed conceptual spaces. In other conceptual spaces, a set of overall weights is assigned to the domains without considering the role. However, here, for two independent sub-concepts, the assigned weights in the same domain might vary.

4. Semantic Inference in Conceptual Spaces

This section aims to design an inference approach in order to provide a semantic characterisation for a new set of unknown observations. The focus of this section is on solving two main questions: (1) how a new (unknown) observation is represented in a conceptual space, and (2) how this representation enables the inference of semantic descriptions for the observation. The first question refers to the problem of *induction* in conceptual spaces theory (Gärdenfors, 2005). To develop such inductions in a conceptual space, it is important to realise which concepts represent a new observed instance. Due to the geometrical representation of the conceptual space, the *similarity* between the instances in the space enables us to define the notion of *inclusion* as an operator to measure the similarity of new observations to the specified concepts within the metric domains. The second question refers to the problem of *symbol grounding* in the conceptual space theory (Aisbett & Gibbon, 2001). The inference of a semantic representation for any input observation in natural language is enabled by defining a *symbol space.*

In general, the proposed inference in a conceptual space consists of the following steps:

- **Defining the *symbol space***, based on the prior knowledge for linguistic characterisation of the concepts and quality dimensions,
- **Inferring *linguistic descriptions***, for each new unknown observation, based on the *inclusion* of its corresponding *instance* in the concepts:
  - **Inference in *conceptual space***: specifying the geometrical location of a new instance within the conceptual space, examining the *inclusion* of the instance, and determining the linguistic labels in the symbol space from the associated concepts and dimensions,
  - **Inference in *symbol space***: annotating and characterising the instance based on the provided set of symbolic terms, and generating linguistic descriptions.

Example 8. Consider the concepts and quality dimensions of the leaf conceptual space in Example 4. A new observed leaf can be either linguistically represented by a known concept (e.g., $C_{\texttt{no}}$) where "*The new observation is a **Nerium** leaf.*", or by a set of related quality dimensions (e.g., $q_{\texttt{el}}$ and $q_{\texttt{ro}}$), such as "*The new observation is an **elongated** and **lance-shaped** leaf.*"

Figure 5 illustrates the step of inferring linguistic descriptions for an unknown observation through the constructed conceptual spaces and its corresponding symbol space. The details of each component are explained in Section 4.2, after formally defining the symbol space in Section 4.1.

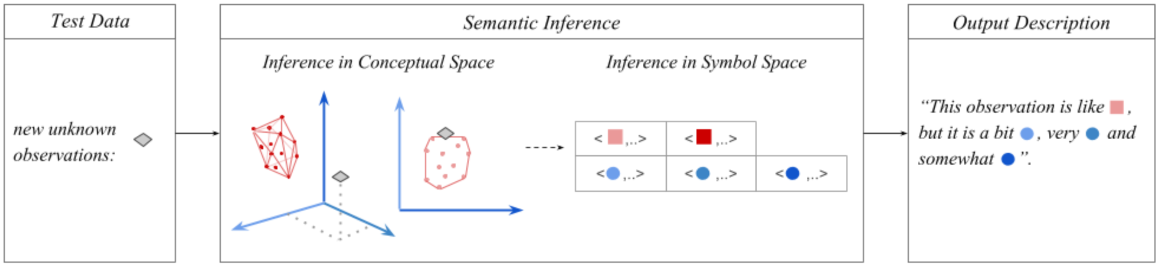

Test Data
new unknown observations:
Semantic Inference
Inference in Conceptual Space
Inference in Symbol Space
< ■ ,...>
< ■ ,...>
< ● ,...>
< ● ,...>
< ● ,...>
Output Description
"This observation is like ■, but it is a bit ●, very ● and somewhat ●".

Figure 5: Illustration of the steps of inferring linguistic descriptions for an unknown observation via the constructed conceptual spaces and its corresponding symbol space. The details of the semantic inference step are explained in Section 4.2.

4.1 Symbol Space Definition

According to a general formulation proposed by Aisbett and Gibbon (2001), a conceptual space can be augmented with a *symbol space*. This extension provides an internal mapping between geometrical elements in conceptual space (such as concepts, dimensions, domains, etc.) and the symbolic labels (typically words) in symbol space.

Definition 6. A symbol space $\mathcal{S}$ of size n is a space containing n *symbol dimensions* $\mathcal{L}^{\mathcal{S}}$, wherein each concept and quality dimension in the conceptual space is linked to a symbol dimension. Symbol dimensions are isomorphic to the real number interval $[0, 1]$.

Based on the definition of Aisbett and Gibbon, the symbol dimensions need to be named by the primitive input labels of the associated concepts (classes) and quality dimensions (features). The construction of the symbol space is a knowledge-based process, wherein the prior knowledge is encoded (Aisbett & Gibbon, 2001). The prior knowledge specifies the symbolic expressions in natural language form, related to the elements of conceptual space S.

We propose a two-layer symbol space containing the symbol dimensions of the concepts, $\mathcal{L}^{\mathcal{C}} = \big[d_{C_1},\ d_{C_2}, \dots\big]$, as *concept layer*, and the symbol dimensions of quality dimensions, $\mathcal{L}^{\mathcal{Q}} = \big[d_{q_1},\ d_{q_2}, \dots\big]$, as *quality layer*. So, for every concept $C_y \in \mathcal{C}$, there is a symbol dimension in the concept layer, and for each quality dimension $q \in \mathcal{Q}$, there is a symbol dimension in the quality layer. Figure 6 shows the associations between the elements in a conceptual space and the two-layer symbol dimensions in a symbol space.

Any instance in a conceptual space is associated with a point in the symbol space, namely a *symbol vector*. For a given instance γ, the associated symbol vector $\mathcal{V}_\gamma$ in $\mathcal{S}$ specifies the applicability of the symbol dimensions for γ in the range 0 to 1 for each dimension (Aisbett & Gibbon, 2001). The symbol vector $\mathcal{V}_\gamma$ is a concatenation of two vectors $\mathcal{V}_\gamma :< \mathcal{V}_\gamma^{\mathcal{C}}, \mathcal{V}_\gamma^{\mathcal{Q}} >$, one vector in the concept layer and one vector in the quality layer, respectively. Thus, $|\mathcal{V}_\gamma^{\mathcal{C}}| = |\mathcal{L}^{\mathcal{C}}|$, and $|\mathcal{V}_\gamma^{\mathcal{Q}}| = |\mathcal{L}^{\mathcal{Q}}|$.

Example 9. Consider the conceptual space of leaves S^l in Example 4. For a new leaf sample as an instance γ, the symbol vector $\mathcal{V}_\gamma$ is defined as a 4-dimensional vector with the concatenation of $\mathcal{V}_\gamma^{\mathcal{C}} =< v_{d_{\texttt{tt}}}, v_{d_{\texttt{no}}} >$, and , $\mathcal{V}_\gamma^{\mathcal{Q}} =< v_{d_{\texttt{el}}}, v_{d_{\texttt{ro}}} >$.

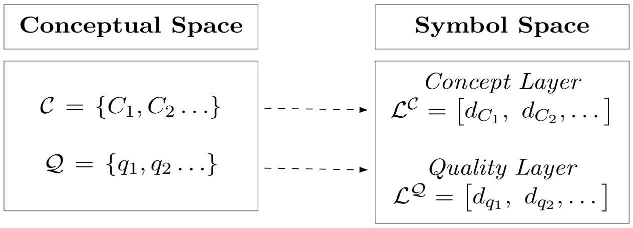

Conceptual Space
Symbol Space
$\mathcal{C} = \{C_1, C_2 \ldots\}$
$\mathcal{Q} = \{q_1, q_2 \ldots\}$
Concept Layer
$\mathcal{L}^{\mathcal{C}} = [d_{C_1},\ d_{C_2}, \ldots]$
Quality Layer
$\mathcal{L}^{\mathcal{Q}} = [d_{q_1},\ d_{q_2}, \ldots]$

Figure 6: Schematic of a conceptual space and the coupled symbol space.

The symbol vector is defined as a two-element vector, wherein each $v_i \in \mathcal{V}_\gamma$ consists of a pair of values $v_i = (\mathit{label}, \mathit{value})$. The *label* shows the related symbolic term, and the *value* shows how representative is the instance to the dimension d_i (either how similar to its concepts or how related to the quality dimensions). The notion $\mathcal{V}^{\mathcal{C}}_\gamma(C_y) = v_{d_{\mathrm{C_y}}}$ indicates the value of the symbol vector in the concept layer for the dimension related to the concept C_y, and similarly, $\mathcal{V}^{\mathcal{Q}}_\gamma(q_j) = v_{d_{\mathrm{q_j}}}$ indicates the value of the symbol vector in the quality layer for the dimension related to the quality dimension q_j. The further sections explain how the elements of a symbol vector for a new instance are assigned values based on the inclusion of the instance within the domains.

4.2 Inferring Linguistic Descriptions for Unknown Observations

For any given unknown observation, the goal is to infer a semantic description in natural language form. The core of the inference process is to cope with the notion of *similarity* in conceptual spaces. In order to place the new instances in the space and choose the best concepts that include (or are similar enough to) an observation (Rickard et al., 2007). The metric structure of a geometrical conceptual space enables the model to measure the semantic similarity of concepts and instances in the space (Adams & Raubal, 2009b). The proposed construction of the conceptual space in Section 3 facilitates these measurements since the representations of concepts and instances span across domains using the geometric elements i.e., convex regions and points. From the point of view of NLG, inferring linguistic descriptions for unknown observations covers the main tasks of an NLG pipeline for generating natural language text out of non-linguistic data: *Content determination*, *Microplanning* (including *lexicalisation*), and *Realisation* (Reiter et al., 2000). This phase employs various developed methods for linguistic descriptions (i.e., fuzzy set theory, see Ramos-Soto et al., 2016) in order to ease the process of quantifying the location of unknown samples within a conceptual space, and infer the proper linguistic terms.

The process of inferring linguistic descriptions for an instance γ' is presented in two following phases.

- *Phase A: Inference in Conceptual Space*, that first determines the inclusion of the new instance γ' in the concepts within the domains $\Delta(\gamma')$ using semantic similarity, and then sets the values of symbol vector $\mathcal{V}_{\gamma'}$, (performing content determination).
- *Phase B: Inference in Symbol Space*, that verbalises the symbol vector $\mathcal{V}_{\gamma'}$ into a set of lexical items which are human-readable descriptions, (performing lexicalisation and realisation).

For a given set of new observations $\mathcal{D}' = \{o'_i : (\mathbf{x}_{o'_i})\}$, let γ' be the corresponding instance to the unknown observation $o' \in \mathcal{D}'$ which is not assigned to any of the known class labels of $\mathcal{Y}$. Also, let $\Delta(\gamma') \subseteq \Delta$ be a set of domains that γ' has corresponding points in each of them[5], where $|\Delta(\gamma')| = k'$. The details of the phases A and B are explained in the following sections.

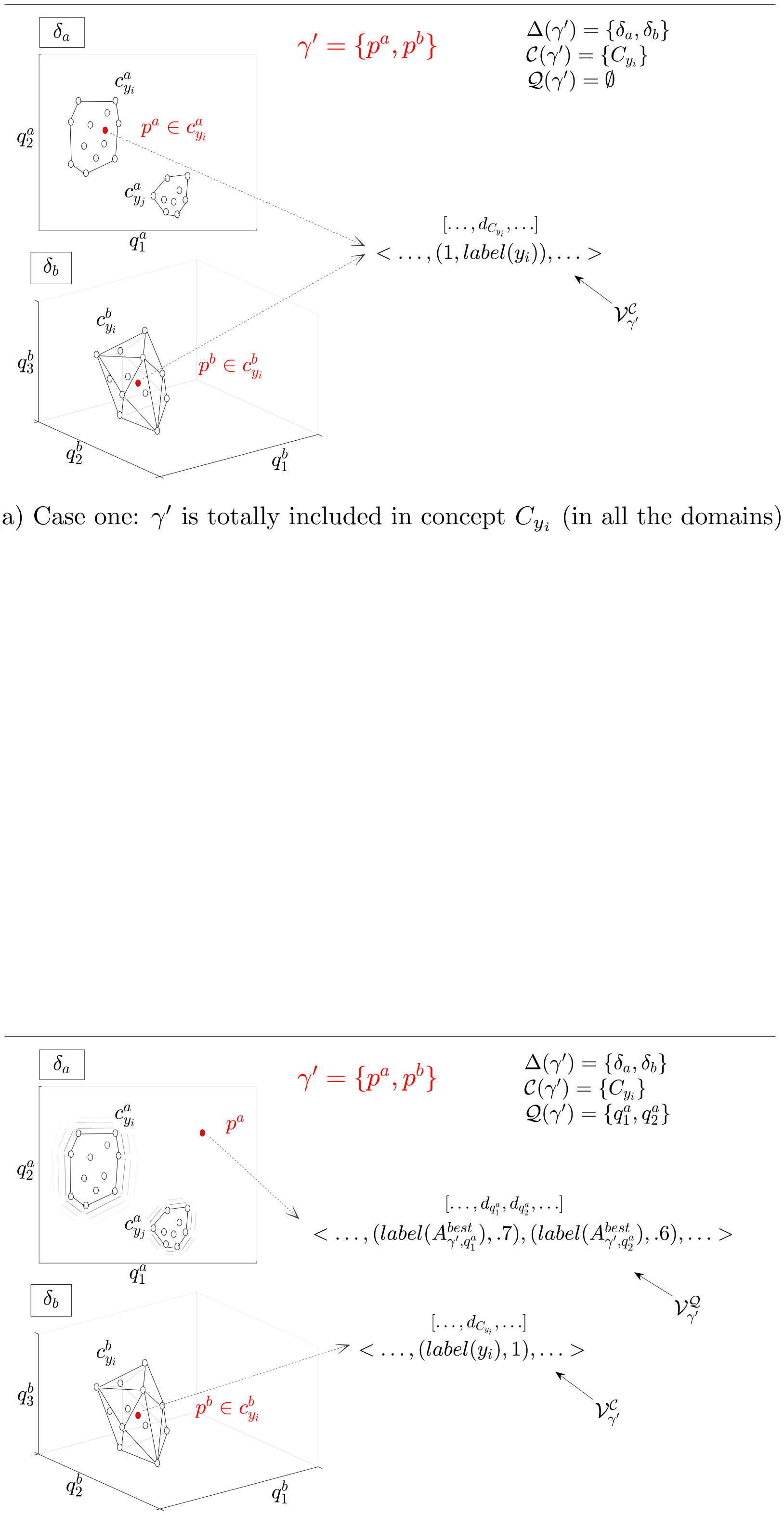
4.2.1 Phase A: Inference in Conceptual Space

In this phase, we first compare the new instance with each of the concepts using the similarity measure to check whether it is included within concept regions or not. Then, based on the result of this inclusion, we initialise a symbol vector and set its values in both concept layer and quality layer. From the NLG perspective, this performs the task of *content determination* (Reiter, 2007), that decides which set of information is required to characterise a new observation in the final description. The process of checking the inclusion is a simple fuzzy extension of an *instance-based* method that measures the membership of the new instance to the nearest labelled region of instances. Although it can be done by any classification approach, we formulate the process with respect to the definitions of the introduced conceptual space.

For an instance γ', the symbol vector $\mathcal{V}_{\gamma'}$ is calculated based on the inclusion of the instance points $p_{\gamma'}$ in different regions within the domains. As defined before, γ' is represented with a set of points $\gamma' = \{p^1_{\gamma'}, \ldots, p^{k'}_{\gamma'}\}$. In general, with placing a new instance in a conceptual space, four cases can occur. Without losing generality, assume γ' consists of two points in two domains δ_a and δ_b, denoted by $\gamma' = \{p^a_{\gamma}, p^b_{\gamma}\}$. Also assume that there are two concepts C_{y_1} and C_{y_2} that have been represented in one or both of these two domains. Figure 7 shows the four different cases concerning various positions of the points p^a and p^b, and their relations to the sub-concepts' regions within the domains. One instance can be located in the space differently as follows:

1. Totally included in a concept within all the domains (case one, Figure 7a),
2. Partially included in just a concept (case two, Figure 7b),
3. Partially included in two distinct concepts (case three, Figure 7c),
4. Not included in any concept (case four, Figure 7d).

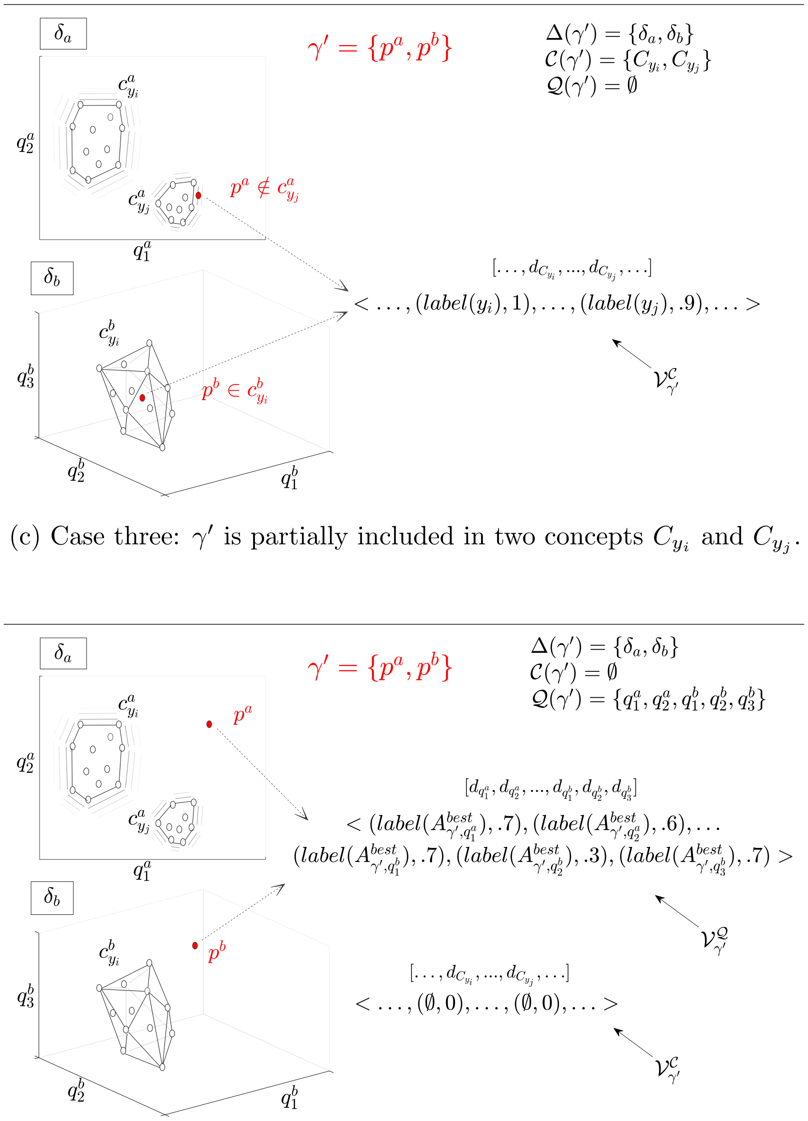
[5] It is notable that a new observation is not necessarily defined in all domains since there might be no calculated values for some of the features/quality dimensions. So, the corresponding instance may not have points in all the provided domains.


δ_a
$\gamma' = \{p^a, p^b\}$
$\Delta(\gamma') = \{\delta_a, \delta_b\}$
$\mathcal{C}(\gamma') = \{C_{y_i}\}$
$\mathcal{Q}(\gamma') = \emptyset$
$c^a_{y_i}$
$p^a \in c^a_{y_i}$
q^a_2
$c^a_{y_j}$
q^a_1
$[\dots, d_{C_{y_i}}, \dots]$
$< \dots, (1, label(y_i)), \dots >$
$\mathcal{V}^{\mathcal{C}}_{\gamma'}$
δ_b
$c^b_{y_i}$
q^b_3
$p^b \in c^b_{y_i}$
q^b_2
q^b_1

(a) Case one: γ' is totally included in concept C_{y_i} (in all the domains).

δ_a
$\gamma' = \{p^a, p^b\}$
$\Delta(\gamma') = \{\delta_a, \delta_b\}$
$\mathcal{C}(\gamma') = \{C_{y_i}\}$
$\mathcal{Q}(\gamma') = \{q^a_1, q^a_2\}$
$c^a_{y_i}$
p^a
q^a_2
$[\dots, d_{q^a_1}, d_{q^a_2}, \dots]$
$< \dots, (label(A^{best}_{\gamma', q^a_1}), .7), (label(A^{best}_{\gamma', q^a_2}), .6), \dots >$
$c^a_{y_j}$
q^a_1
$\mathcal{V}^{\mathcal{Q}}_{\gamma'}$
δ_b
$[\dots, d_{C_{y_i}}, \dots]$
$< \dots, (label(y_i), 1), \dots >$
$c^b_{y_i}$
q^b_3
$p^b \in c^b_{y_i}$
$\mathcal{V}^{\mathcal{C}}_{\gamma'}$
q^b_2
q^b_1

(b) Case two: γ' is partially included in concept C_{y_i} (in one domain).


δ_a
$\gamma' = \{p^a, p^b\}$
$\Delta(\gamma') = \{\delta_a, \delta_b\}$
$\mathcal{C}(\gamma') = \{C_{y_i}, C_{y_j}\}$
$\mathcal{Q}(\gamma') = \emptyset$
$c^a_{y_i}$
q^a_2
$c^a_{y_j}$
$p^a \notin c^a_{y_j}$
q^a_1
δ_b
$[\ldots, d_{C_{y_i}}, ..., d_{C_{y_j}}, \ldots]$
$< \ldots, (label(y_i), 1), \ldots, (label(y_j), .9), \ldots >$
$c^b_{y_i}$
q^b_3
$p^b \in c^b_{y_i}$
$\mathcal{V}^{\mathcal{C}}_{\gamma'}$
q^b_2
q^b_1
δ_a
$\gamma' = \{p^a, p^b\}$
$\Delta(\gamma') = \{\delta_a, \delta_b\}$
$\mathcal{C}(\gamma') = \emptyset$
$\mathcal{Q}(\gamma') = \{q^a_1, q^a_2, q^b_1, q^b_2, q^b_3\}$
$c^a_{y_i}$
p^a
q^a_2
$[d_{q^a_1}, d_{q^a_2}, ..., d_{q^b_1}, d_{q^b_2}, d_{q^b_3}]$
$< (label(A^{best}_{\gamma',q^a_1}), .7), (label(A^{best}_{\gamma',q^a_2}), .6), \ldots$
$(label(A^{best}_{\gamma',q^b_1}), .7), (label(A^{best}_{\gamma',q^b_2}), .3), (label(A^{best}_{\gamma',q^b_3}), .7) >$
$c^a_{y_j}$
q^a_1
δ_b
$\mathcal{V}^{\mathcal{Q}}_{\gamma'}$
$c^b_{y_i}$
p^b
$[\ldots, d_{C_{y_i}}, ..., d_{C_{y_j}}, \ldots]$
$< \ldots, (\emptyset, 0), \ldots, (\emptyset, 0), \ldots >$
q^b_3
$\mathcal{V}^{\mathcal{C}}_{\gamma'}$
q^b_2
q^b_1

(c) Case three: γ' is partially included in two concepts C_{y_i} and C_{y_j}.

(d) Case four: γ' is not included in any concept.

Figure 7: An illustration of four different cases concerning the various positions of an instance points $\gamma' = \{p^a, p^b\}$ within two domains δ_a and δ_b, along with the assigned values to symbol vector $\mathcal{V}_{\gamma'}$, according to the inclusion of the points in the presented sub-concepts.

In order to assign a concept to γ', it is necessary to check the inclusion of the instance points in the concept's regions within $\Delta(\gamma')$ using a similarity measure. Within a single domain, two states need to be considered: 1) If the instance point is included in a region, then the region's concept will be assigned to γ'. So, the corresponding symbol dimension of the concept will be activated in the symbol vector of γ' (in the concept layer). 2) If there is no region that the instance belongs to, then no concept will be assigned to γ' within that domain. The symbol dimensions related to the quality dimensions of the domain will be activated in the symbol vector of γ' (in the quality layer). Formally, the symbol vector $\mathcal{V}_{\gamma'}$

Algorithm 4: Inference in Conceptual Space (set symbol vectors)

```
Function ConceptualSpaceInference($(\gamma', \mathcal{Q}, \Delta, \mathcal{C})$)
    foreach $\delta_i \in \Delta(\gamma')$ do
        $p \leftarrow p^i_{\gamma'} \in \gamma'$
        // Set the symbol vector in concept layer
        foreach $c \in \delta_i$ do
            $sim_{\gamma',C_y} = max(sim_{\gamma',C_y}, \mathcal{G}(p, c))$                         // $C_y \ni c$
            if $sim_{\gamma',C_y} \neq 0$                        // cases 1 and 3 (and partially 2)
            then
                $label_{\gamma',C_y} = label(C_y)$
            else
                $label_{\gamma',C_y} = \emptyset$
            $\mathcal{V}^{\mathcal{C}}_{\gamma'}(C_y) = (label_{\gamma',C_y}, sim_{\gamma',C_y})$
        // Set the symbol vector in quality layer
        if $\mathcal{V}^{\mathcal{C}}_{\gamma'}(C_y) == (\emptyset, 0)$                         // case 4 (and partially 2)
        then
            foreach $q \in \delta_i$ do
                $degree_{\gamma',q} = \mathcal{H}(p, q)$
                $label_{\gamma',q} = label(A^{best}_{\gamma',q})$
                $\mathcal{V}^{\mathcal{Q}}_{\gamma'}(q^i) \leftarrow (label_{\gamma',q}, degree_{\gamma',q})$
    return $\mathcal{V}_{\gamma'} : \langle\ \mathcal{V}^{\mathcal{C}}_{\gamma'}, \mathcal{V}^{\mathcal{Q}}_{\gamma'}\ \rangle$
```

gets the values in each domain $\delta_i \in \Delta(\gamma')$ as follows: We first define a *Graded Membership* function, $\mathcal{G}(p^i_{\gamma'}, c^i)$, as an inclusion operator to determine the similarity degree of a point $p^i_{\gamma'}$ to the region of a sub-concept $c^i \in \delta_i$ within δ_i. If $p^i_{\gamma'}$ is similar enough to the convex region of the sub-concept with a certain membership degree, then the value of $\mathcal{G}(p^i_{\gamma'}, c^i)$ is set to $\mathcal{V}^{\mathcal{C}}_{\gamma'}(C_y)$, where $C_y \ni c^i$. Moreover, we define a *Graded Quality* function, $\mathcal{H}(p^i_{\gamma'}, q^i)$, to measure to what degree $p^i_{\gamma'}$ belongs to a quality dimension $q^i \in \mathcal{Q}(\delta_i)$. If $p^i_{\gamma'}$ is not included in any of the sub-concepts, then the value of $\mathcal{H}(p^i_{\gamma'}, q^i)$ is set to $\mathcal{V}^{\mathcal{Q}}_{\gamma'}(q^i)$.

Algorithm 4 shows the steps to set the symbol vector values while iterating through all the involved domains $\Delta(\gamma')$. Both graded membership function and graded quality function are formally defined in the following sections.

Graded Membership Function

The problem of *inclusion* has been studied in the literature of the conceptual spaces theory with various definitions such as inclusion operator (Adams & Raubal, 2009b), graded similarity (Gärdenfors & Williams, 2001), and graded membership (Hampton, 2007; Decock & Douven, 2014), which calculate the similarity of the instance points to the regions based on their geometrical distances, with or without considering the gradedness of membership. Here, since the convex regions of concepts are constructed based on the observed instances, it does not make sense to rigidly adhere to the *crisp* boundaries of the calculated regions.

So, a point which is not certainly inside the calculated boundaries a region, but is similar enough to the region, can be counted as a member of the region's concept.

The *Graded Membership function* $\mathcal{G}(p, c)$ is defined as an inclusion operator between a given point p and a defined sub-concept $c : \langle \eta_c, \phi_c \rangle$. This function shows how similar is p to the convex region η_c of c with the certain set of weights ϕ_c within a metric domain δ. The graded membership is calculated by applying geometrical algorithms that consider whether an n-dimensional point is included in the n-dimensional convex hull or not. If point p is included in the region η_c, then $\mathcal{G}(p, c)$ is equal to one. But if p is outside the region, then the similarity of point p to the region is defined as a monotonic decreasing function (Decock & Douven, 2014) which is measured using a fuzzy membership function of distance $sim(p, c) = f[d(p, c)]$. This similarity takes values between $[0, 1]$ and expresses the graded degree of inclusion of p in c. From the experiments on similarity cognition, the similarity can be measured as an exponential decay function of the distance: $sim(d) = e^{-rd}$ (Shepard et al., 1987) (where r is a constant factor). Using a fuzzy membership function to measure the similarity has the advantage of using notions from the fuzzy set theory that provide linguistic descriptions for the fuzzy output degrees (Rickard, 2006).

Several methods have been proposed to compute the distance of a point p to a convex region η_c. The *Hausdorff distance* $d^H(p, \eta_c)$ (Aisbett & Gibbon, 2001) relies on the definition of a distance measure between two n-dimensional points (namely *Weighted Minkowski Metric*). As a function of the similarity measure, we define a graded membership function, inspired by Hampton's definition (Hampton, 2007) in which a determinate boundary region of membership is assumed. For the points inside the region, the membership value is one. For the points out of the region, with a given lower-bound threshold θ_L for each concept's region, if $d(p, \eta_c) \leq \theta_L$, then p is similar enough to be counted as a member of c, and if $d(p, \eta_c) > \theta_L$, then p is far to be counted as an instance of c (Rickard, 2006)[6].

Definition 7. The graded membership function $\mathcal{G} : \delta \to [0, 1]$ is the similarity measure between a point p and a sub-concept $c \in C_y$ as:

$$\mathcal{G}(p, c) = \begin{cases} 1 & if\ p \in \eta_c \\ e^{-rd^H(p, \eta_c)} & if\ p \notin \eta_c\ \&\ d^H(p, \eta_c) \leq \theta_L \\ 0 & if\ p \notin \eta_c\ \&\ d^H(p, \eta_c) > \theta_L \end{cases} \tag{6}$$

The symbol vector of a new instance in the concept layer ($\mathcal{V}^{\mathcal{C}}_{\gamma'}$) is set by the graded membership function by measuring the similarity of an instance point to each of the sub-concept's regions within the domain (Algorithm 4). As we mentioned in 4.1, each $v_i \in \mathcal{V}_\gamma$ consists of a pair of values $v_i = (label, value)$. The similarity values greater than zero will lead to assigning a non-empty label to the corresponding concept's index in symbol vector. Formally, for a given γ' and C_y, two elements $(label, value)$ are calculated as: $\mathcal{V}^{\mathcal{C}}_{\gamma'}(C_y) = (label_{\gamma', C_y}, sim_{\gamma', C_y})$, where

$$sim_{\gamma', C_y} = max_{p_i \in \gamma', c_j \in C_y}(\mathcal{G}(p_i, c_j)), \tag{7}$$

[6] The usual definition of graded membership function (Hampton, 2007; Decock & Douven, 2014) is slightly different, where it is based on three thresholds to define the lower, upper, and the middle level of the boundary regions. However, similar to these studies, the lower-bound threshold varies for each sub-concept's region depending on the density and the size of the region.

and

$$label_{\gamma',C_y} = \begin{cases} label(C_y) & if\ sim_{\gamma,C_y} > 0 \\ \emptyset & o.w \end{cases} \tag{8}$$

Example 10. Figs. 7a, 7b, and 7c show the example values of the graded membership function ($\mathcal{G}$) calculated for the points p^a and p^b based on their positions and distances to the convex regions of the sub-concepts in the space. For example in Figure 7c, suppose in δ_a, $\mathcal{G}(p^a_{\gamma'}, c^a_{y_j}) = 0.9$. Then, the elements of $\mathcal{V}^{\mathcal{C}}_{\gamma'}(C_{y_j})$ are set to $(label(C_{y_j}),\ 0.9)$.

Graded Quality Function

According to Algorithm 4, if a point p in a domain δ is not similar enough to any sub-concept within δ, then the values of the symbol vector in the quality layer will be set based on the graded value of p for each quality dimension of δ. Recall from Section 3, a quality dimension $q : \langle H_q, I_q, \mu_q \rangle$ contains a family of membership functions μ_q, representing the linguistic terms related to graded values of q. In particular, this function is defined as a fuzzy granulation in order to exploit the linguistic characterisation of feature values, which are identified by prior knowledge. To formalise μ_q, we define fuzzy membership functions for a set of pre-defined label classes which forms a *fuzzy partition* of the interval I_q (Novák, 2016). Suppose A_i is a class (label) acquired for q (e.g., linguistic label *tall* for feature *height*). The corresponding fuzzy set is defined as: $A_i = \{(x, \mu_{A_i}) \mid x \in I_q\}$, where μ_{A_i} is a sigmoidal membership function with certain parameters to define the lower and upper boundaries of the function.

Example 11. Consider the *elongation*, described in Example 2. A set of label classes to describe the elongation is { $A_{\texttt{cir}}$ =‘*circular*’, $A_{\texttt{elp}}$ =‘*elliptical*’, and $A_{\texttt{eld}}$ =‘*elongated*’}. Then the family of membership functions for $q_{\texttt{el}}$ is $\mu_{q_{\texttt{el}}} = \{\mu_{A_{\texttt{cir}}}, \mu_{A_{\texttt{elp}}}, \mu_{A_{\texttt{eld}}}\}$.

This linguistic mapping grounds a symbolic representation of numeric interval values of the quality dimensions. The graded quality value of an instance γ' for a quality dimension q is calculated based on the quality dimension value of the instance point as $p_q = p_{\gamma'}(q)$, where $p_q \in I_q$. Using the defined fuzzy membership functions, we map p_q into the fuzzy set best matching the given value. Using the functions in μ, we are able to set the values of the symbol vector in the quality layer. Recall $p = \langle p_{q_1}, \ldots, p_{q_{|Q(\delta)|}} \rangle$ as the vector of quality dimension values for the point p in δ.

Definition 8. Graded quality function $\mathcal{H} : I_q \to [0, 1]$ is the degree of membership, wherein for a quality dimension q, it returns the maximum degree of membership of p_q using the membership functions in μ_q, as:

$$\mathcal{H}(p, q) = max_{\mu_{A_i} \in \mu_q}\ \mu_{A_i}(p_q) \tag{9}$$

The symbol vector of a new instance in the quality layer ($\mathcal{V}^{\mathcal{Q}}_{\gamma'}$) is filled with the values of the graded quality function (Algorithm 4). Similar to the concept layer, each $v_i \in \mathcal{V}_\gamma$ consists of a pair of values $v_i = (label, value)$. The value of the graded quality function, which is the maximum degree of membership of v_i, assigns the best match fuzzy subset (i.e., symbolic label) to the corresponding quality dimension's index in symbol vector. Formally, for a given

Algorithm 5: Inference in Symbol Space (set linguistic terms)

```
Function $SymbolSpaceInference(\mathcal{V}_{\gamma'})$
    // Annotation in the concept layer
    foreach $C_y \in \mathcal{C}$ do
        if $\mathcal{V}_{\gamma',\mathcal{C}}(C_y) \neq (\emptyset, 0)$ then
            $\mathcal{C}(\gamma') \leftarrow \mathcal{C}(\gamma') \cup \{C_y\}$
    $\mathcal{O}_{\mathcal{C}}(\gamma') \leftarrow OrderConceptLabels(\mathcal{C}(\gamma'))$
    $\mathcal{T}_{\mathcal{C}}(\gamma') \leftarrow Annotate(\mathcal{O}_{\mathcal{C}}(\gamma'), \mathcal{V}_{\gamma',\mathcal{C}})$          // in concept layer
    // Characterisation in the quality layer
    foreach $\delta \notin \Delta(\mathcal{C}(\gamma'))$ do
        $\mathcal{Q}(\gamma') \leftarrow \mathcal{Q}(\gamma') \cup \{\mathcal{Q}(\delta)\}$
    $\mathcal{O}_{\mathcal{Q}}(\gamma') \leftarrow OrderQualityLabels(\mathcal{Q}(\gamma'))$ $\mathcal{T}_{\mathcal{Q}}(\gamma') \leftarrow Characterise(\mathcal{Q}(\gamma'), \mathcal{V}_{\gamma',\mathcal{Q}})$      // in
    quality layer
    // Linguistic Realisation
    $\mathcal{T}_{\gamma'} \leftarrow Realise(\mathcal{T}_{\mathcal{C}}(\gamma'), \mathcal{T}_{\mathcal{Q}}(\gamma'))$
return $\mathcal{T}_{\gamma'}$
```

γ' and C_y, two elements $(label, value)$ of are calculated as: $\mathcal{V}^{\mathcal{Q}}_{\gamma'}(q) = (label_{\gamma',q}, degree_{\gamma',q})$, where

$$degree_{\gamma',q} = \mathcal{H}(p, q), \tag{10}$$

and

$$label_{\gamma',q} = label(A^{best}_{\gamma',q}). \tag{11}$$

Here, $A^{best}_{\gamma',q}$ is the fuzzy subset with the maximum degree of membership such that $\mu_{A^{best}_{\gamma',q}} \in \mu_q$ and $\forall(\mu_{A_i} \in \mu_q) : \mu_{A^{best}_{\gamma',q}}(p_q) \geq \mu_{A_i}(p_q)$.

Example 12. Figs. 7b and 7d show the example values of the graded quality function ($\mathcal{H}$) calculated for the points p^a and p^b based on their values related to the quality dimensions of the two domains. For example in Figure 7b, suppose p^a is not included in any region in δ_a. Then, two vector indices of the symbol vector in the quality layer get values as: $\mathcal{V}^{\mathcal{Q}}_{\gamma'}(q^a_1) = (label(A^{best}_{\gamma',q^a_1}), 0.7)$ and $\mathcal{V}^{\mathcal{Q}}_{\gamma'}(q^a_2) = (label(A^{best}_{\gamma',q^a_2}), 0.6)$.

4.2.2 Phase B: Inference in Symbol Space

In this phase, the aim is to infer linguistic descriptions from the symbol vector of an unknown instance, which is to identify lexical items that suitably map to the symbol vector elements. From the NLG perspective, this is the task of *lexicalisation*, that decides which linguistic terms (i.e., natural words) should be selected from the determined content (Reiter, 2007). This task can be done by verbalising the linguistic labels that are calculated and stored in the symbol vector. The verbalisation is done either with *annotating* a new instance via the concept labels or with *characterising* the instance via the quality dimension labels. The tasks of *annotation* and *characterisation* will assign a set of lexical items to an unknown observation. This set of linguistic terms is then turned to the natural language phrases (i.e., sentences) using the *realisation* tools in NLG systems. Algorithm 5 shows the steps of the tasks in phase B.

Annotation in the Concept Layer

Annotation for an instance γ' is to annotate a set of linguistic labels which are derived from the associated concepts in the concept layer of symbol vector $\mathcal{V}_{\gamma',\mathcal{C}}$. Each C_y is included in the set of associated concepts of the instance, $\mathcal{C}(\gamma')$, if the corresponding element in $\mathcal{V}_{\gamma',\mathcal{C}}$ is not empty. Formally, $C_y \in \mathcal{C}(\gamma') \iff \mathcal{V}_{\gamma',\mathcal{C}}(C_y) \neq (\emptyset, 0)$.

Example 13. Considering Figs. 7a and 7b, γ' is associated with only one concept C_{y_i}. So, $\mathcal{C}(\gamma') = \{C_{y_i}\}$. In Figure 7c, γ' is associated with two concepts C_{y_i} and C_{y_j}. So, $\mathcal{C}(\gamma') = \{C_{y_i}, C_{y_j}\}$. Finally, in Figure 7d, there are no associated concepts. So, $\mathcal{C}(\gamma') = \emptyset$.

After determining $\mathcal{C}(\gamma')$, if γ' is associated with two or more distinct concepts as $\mathcal{C}(\gamma') = \{C_{y_i}, C_{y_j}, \ \ldots\}$, then γ' is an instance of all the associated concepts. In this case, an extra process is needed to sort and combine the concept labels to annotate γ' with a new set of linguistic labels. The task of *concept combination* is discussed widely in the literature of conceptual space theory (Rickard, 2006; Gärdenfors, 2000; Lewis & Lawry, 2016). What is important for our inference is to distinguish which concept labels are the *modifiers* and which are the *modified* concepts. This distinction leads to order the labels in the final linguistic expression (Adams & Raubal, 2009b). In particular, we define an *ordered set*[7] of the associated concepts, $\mathcal{O}_{\mathcal{C}}(\gamma') = \{C'_1, C'_2, \ \ldots\}$ to prioritise the modifier concepts over modified ones. Since there is no background knowledge to define the semantic order of the associated concepts, the ordering process relies based on the *graded membership* values that can be retrieved from $\mathcal{V}_{\gamma',\mathcal{C}}$. The set of annotations for γ' then is defined as an ordered set of lexical items $\mathcal{T}_{\mathcal{C}}(\gamma') = \{label(C'_1), label(C'_2), \ldots\}$. These annotations are the corresponding linguistic terms to the ordered concepts in $\mathcal{O}_{\mathcal{C}}(\gamma')$, which are retrieved from the labels in $\mathcal{V}_{\gamma',\mathcal{C}}$ for the concepts in $\mathcal{C}(\gamma')$.

Example 14. For the conceptual space of leaves presented in Example 4, suppose an unknown leaf sample γ' is associated with both known concepts *Tilia* and *Nerium.* Assume that $\mathcal{V}_{\gamma',\mathcal{C}}(C_{\texttt{tt}}) = (label(C_{\texttt{tt}}), 0.5)$ and $\mathcal{V}_{\gamma',\mathcal{C}}(C_{\texttt{no}}) = (label(C_{\texttt{no}}), 0.9)$. Then $\mathcal{T}_{\mathcal{C}}(\gamma') = \{label(C_{\texttt{no}}) = \text{‘}\textit{Nerium}\text{’}, label(C_{\texttt{tt}}) = \text{‘}\textit{somewhat Tilia}\text{’}\}$.

Characterisation in the Quality Layer

Characterisation for γ' is to assign the linguistic descriptions of the associated quality dimension based on the values in the quality layer of the symbol vector $\mathcal{V}_{\gamma',\mathcal{Q}}$. The motivation behind the characterisation comes from the lack of the concept annotation in the cases with no associated concept within the domains (like case four and partially case two in Figure 7). This is especially important for those instances that are completely unknown for the systems and are not representable by any of the defined concepts, but still are explainable with their quality dimensions' values.

According to Algorithm 4, if γ' within a domain does not belong to any sub-concept, then $\mathcal{V}_{\gamma',\mathcal{Q}}$ gets values from the domain's quality dimensions. Obviously, if γ' has even one associated concept within the domain, there is no need to involve the quality dimensions of that domain in the characterisation process. For the calculated $\mathcal{V}_{\gamma',\mathcal{Q}}$, each quality dimension

[7] In the set theory, an ordered set is defined as a set of elements, plus a relation $\leq$ between each pair of elements, that presents the order of them (Weisstein, 2016).

q is included in the set of associated quality dimensions $\mathcal{Q}(\gamma')$, if the corresponding elements in the $\mathcal{V}_{\gamma',\mathcal{Q}}$ are not empty. Formally, $q \in \mathcal{Q}(\gamma') \iff \mathcal{V}_{\gamma',\mathcal{Q}}(q) \neq (\emptyset, 0)$.

Example 15. Considering Figure 7b, γ' is not associated with any concepts within δ_a. So, $\mathcal{Q}(\gamma') = \{q_1^a, q_2^a\}$. Also, in Figure 7d, there are no associated concepts in any of the domains. So, $\mathcal{Q}(\gamma') = \{q_1^a, q_2^a, q_1^b, q_2^b, q_3^b\}$. In Figs. 7a and 7c, $\mathcal{Q}(\gamma') = \emptyset$.

Similar to the process of annotation, a sorting operation is needed to derive the order of quality dimension labels in the final linguistic descriptions. Relying on the *graded quality* values in $\mathcal{V}_{\gamma',\mathcal{Q}}$, we define an ordered set of the associated quality dimensions, $\mathcal{O}_\mathcal{Q}(\gamma') = \{q'_1, q'_2, \ \ldots\}$. The characterisation set for γ' then is simply defined as an ordered set of lexical items $\mathcal{T}_\mathcal{Q}(\gamma') = \{label(A^{best}_{\gamma',q'_1}),\ label(A^{best}_{\gamma',q'_2}),\ \ldots\}$. These characterisations are retrieved from the corresponding labels in $\mathcal{V}_{\gamma'.\mathcal{Q}}$, for the quality dimensions in $\mathcal{Q}(\gamma')$.

Example 16. Considering Example 15, suppose γ' is not associated with any known concepts. Assume that for the quality dimensions *elongation* and *roundness*, $\mathcal{V}_{\gamma',\mathcal{Q}}(q_{\texttt{el}}) = (label(A_{\texttt{eld}}), 0.9)$ and $\mathcal{V}_{\gamma',\mathcal{Q}}(q_{\texttt{ro}}) = (label(A_{\texttt{ro}}), 0.7)$ (referring to Example 11). Then, $\mathcal{T}_\mathcal{Q}(\gamma')$ $= \{label(A_{\texttt{eld}}) =$ '*elongated*', $label(A_{\texttt{ro}}) =$ '*lanced shape*'$\}$.

Linguistic Realisation

Linguistic realisation is the process of applying a set of rules to abstract representations of the lexical items in order to specify well-formed sentences in natural language, which are syntactically and morphologically correct. Some of the linguistic realisations represent sentences by *template-like structures* when only limited syntactic variability is needed in the output description (Reiter & Dale, 1997). One instance of an output format for the sentences to describe the observations is as follows:

"*This* [*Obs.*] [*be/be not/be like*] [*a* ***con.label1***] [*and* [*a* ***con.label2***] *and* ...],
[*but/also*] *it* [*be/have*] [***dim.label1***] [*and/with* [***dim.label2***] *and/with* ...]."

As this descriptive sentence is linguistically formed in a simple format, applying any realisation technique (e.g., *SimpleNLG* engine, see Gatt & Reiter, 2009) will produce grammatically correct sentences as the output text.

Example 17. Consider the set of annotations and characterisations from Examples 14 and 16. Then, the output of realisation will be a message like:
"*This unknown leaf observation is like* ***Nerium*** *leaves and somewhat* ***Tilia*** *leaves, but it is an* ***elongated*** *and a* ***lance-shaped*** *leaf.*"

5. Case Studies

This section presents a validation of the formal methods described in the previous sections. Two case studies are investigated to show the feasibility and the plausibility of the proposed approach in real-world applications. For both case studies, the primitive set of features is initialised by expert-oriented questionnaires or domain-oriented background knowledge. These semantically interpretable features are described in natural language and are able to distinguish the known classes from each other perceptually.

5.1 Leaf Data Set

The leaf data set (Silva et al., 2013) is a set of photographed leaf samples from different plant species. Here, we selected six species as the labelled data set (See Figure 8), and the rest of the species are used as unlabelled data. The leaf data set is a good first example, as it provides a tangible example of physical objects while the vocabulary used to describe the leaves is not necessarily familiar to non-specialists.

5.1.1 Constructing a Conceptual Space of Leaves

The leaf data set includes 72 known leaf samples that are categorised in six species. Formally, $\mathcal{D}^l = \{o_1, \ldots, o_{72}\}$ and $\mathcal{Y}^l = \{\ y_{\texttt{sa}}$:'`Salix Atrocinera`', $y_{\texttt{qr}}$:'`Quercus Robur`', $y_{\texttt{ia}}$:'`Ilex Aquifolium`', $y_{\texttt{no}}$:'`Nerium Oleander`', $y_{\texttt{tt}}$:'`Tilia Tomentosa`', $y_{\texttt{ap}}$:'`Acer Palmatum`' $\}$. Figure 8 shows the prototypical samples of leaf species for the leaf labels in $\mathcal{Y}$, along with their popular names. [8] According to the set of class labels $\mathcal{Y}^l$, the set of concepts is defined as $\mathcal{C}^l = \{C_{\texttt{ia}}, C_{\texttt{tt}}, C_{\texttt{no}}, C_{\texttt{qr}}, C_{\texttt{ap}}, C_{\texttt{sa}}\}$. Here, the concepts are initiated, but the representation of each concept will be formally presented later.

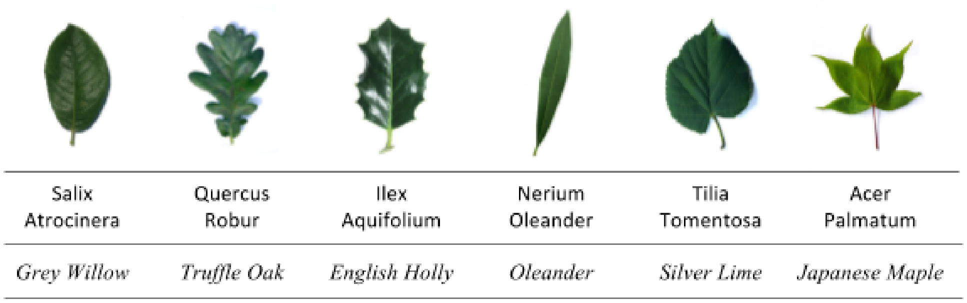


Figure 8: Six species as the known leaves in the leaf data set. The first row of labels shows the scientific names (Silva et al., 2013), and the second row of labels shows the common names (Wu et al., 2007).

The first step to build the conceptual space of leaves is to specify the initial set of features that characterise the leaves. The main criterion while initialising the features is how descriptive or interpretable the chosen features are in the linguistic form. In other words, we are looking for such features that are representable in the human natural language with a perceptual interpretation. For example, the values of the *area* and the *perimeter* features might be useful for statistical analysis or classification tasks, but these features do not carry meaningful information to describe and distinguish the leaf observations. In contrast, a feature like *elongation* meaningfully describes a perceptual feature of a leaf observation.

Besides the leaf samples in different species, Silva et. al. (2013) have provided a set of attributes that describe the shape and texture features of leaves. Among the semantic

[8] Note that in the model, we applied the scientific names of the leaves as labels that have been used in the original data set. However, in the final descriptions for the evaluation, we used the common names of leaves (Wu, Bao, Xu, Wang, Chang, & Xiang, 2007) which were more familiar to the general users.

attributes, the following features are used in our model as the initial set of features $\mathcal{F}^l = \{X_i : \langle H_{X_i}, I_{X_i}\rangle\}$:
$X_{\texttt{ec}}$: $\langle$‘`Eccentricity`’, $[0,1]\rangle$ (eccentricity of the ellipse),
$X_{\texttt{ar}}$: $\langle$‘`Aspect Ratio`’, $[1,\inf)\rangle$ (values close to 1 indicate an elongated shape),
$X_{\texttt{el}}$: $\langle$‘`Elongation`’, $[0,1]\rangle$ (minimum is achieved for a circular region),
$X_{\texttt{so}}$: $\langle$‘`Solidity`’, $(0,1)\rangle$, (how well the leaf fits a convex shape),
$X_{\texttt{if}}$: $\langle$‘`Isoperimetric Factor`’, $[1,\inf)\rangle$ (curvy intertwined contours yield low values),
$X_{\texttt{lo}}$: $\langle$‘`Lobedness`’, $(0,\inf)\rangle$ (characterises how lobed a leaf is),
$X_{\texttt{mi}}$: $\langle$‘`Maximal Indentation Depth`’, $(0,1)\rangle$, (how deep are the indentations),
$X_{\texttt{sc}}$: $\langle$‘`Stochastic Convexity`’, $[0,1]\rangle$, (probability of a random segment in a leaf to be fully contained).

Domain Specification for Leaf Data Set The values of these features for every observation are acquired from the work of Silva et. al. (2013). After that, we have applied the conceptual space construction approach with the inputs of labelled observations $\mathcal{D}^l$, label set $\mathcal{Y}^l$, and feature set $\mathcal{F}^l$. The algorithm first applies the feature filtering approach, i.e. MIFS (Algorithm 1) to provide a ranking matrix which shows the mutual relations of features and labels. Then, using Algorithm 2, a feature subset grouping is performed. Figure 9 illustrates the created bipartite graph, which leads to determining the domains and quality dimensions.

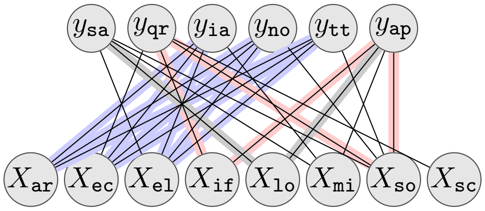


Figure 9: The bipartite graph, presenting the relevance of the features and the labels in the leaf data set. Also, three chosen bicliques (as the domains) are highlighted with blue, red, and grey edges.

The chosen bicliques (with the highest scores) determine the three domains $\Delta^l = \{\delta_1, \delta_2, \delta_3\}$, where each domain is specified as follows:

- Domain $\delta_1 = \langle \mathcal{Q}(\delta_1), \mathcal{C}(\delta_1), \omega_{\delta_1}\rangle$, wherein $\mathcal{Q}(\delta_1) = \{q_{\texttt{ar}},\ q_{\texttt{el}},\ q_{\texttt{ec}}\}$ and $\mathcal{C}(\delta_1) = \{C_{\texttt{ia}}, C_{\texttt{tt}}, C_{\texttt{no}}\}$.
- Domain $\delta_2 = \langle \mathcal{Q}(\delta_2), \mathcal{C}(\delta_2), \omega_{\delta_2}\rangle$, wherein $\mathcal{Q}(\delta_2) = \{q_{\texttt{so}},\ q_{\texttt{if}}\}$ and $\mathcal{C}(\delta_2) = \{C_{\texttt{qr}}, C_{\texttt{ap}}\}$.
- Domain $\delta_3 = \langle Q(\delta_3), C(\delta_3), \omega_{\delta_3}\rangle$, wherein $\mathcal{Q}(\delta_3) = \{q_{\texttt{lo}}\}$ and $\mathcal{C}(\delta_3) = \{C_{\texttt{sa}}, C_{\texttt{ap}}\}$.

Figure 10 depicts a graphical presentation of the determined domains with the corresponding quality dimensions and concepts. As an example, domain δ_2 is specified by two quality dimensions ‘`solidity`’ and ‘`isoperimetric factor`’, and is associated with two concepts ‘`Quercus`’ and ‘`Acer`’. An example of calculated weights in a domain is $\omega_{\delta_2}(C_{\texttt{ap}}, q_{\texttt{so}}) = 0.61$, which shows the salience of the relation between leaf concept ‘`Acer`’ and quality dimension ‘`solidity`’ within δ_2. Although the process of deriving the domains is data-driven, there may be an interpretation for each specified domain. For instance, one

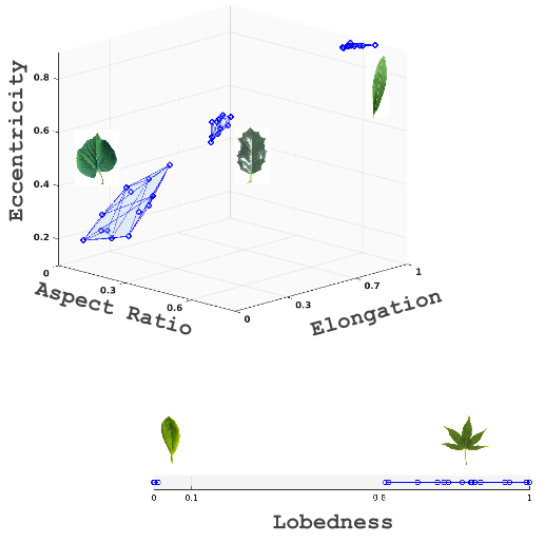


Figure 10: The conceptual space of leaf data set: a graphical presentation of the determined domains with the corresponding quality dimensions and concepts.

can say that δ_1 illustrates the *convexity* of the known leaves, while δ_2 shows the *indentation* of the known leaves (see Figure 10).

As an output of the domain specification phase for the conceptual space of leaves, the set of quality dimensions is $\mathcal{Q}^l = \{q_{\mathtt{ar}},\ q_{\mathtt{el}},\ q_{\mathtt{ec}},\ q_{\mathtt{so}},\ q_{\mathtt{if}},\ q_{\mathtt{lo}}\}$, and the set of instances is $\Gamma^l = \cup_{y \in \mathcal{Y}}\ \Gamma(y)$, where $|\Gamma^l| = |\mathcal{D}^l|$. Each $\gamma \in \Gamma^l$ corresponds to a known leaf sample $o \in \mathcal{D}^l$, and consists of three points (one in each domain).

Concept Representation for Leaf Concepts According to the output of concept representation, each leaf concept in $\mathcal{C}^l$ appears in only one domain and thus has exactly one sub-concept, except concept C_{ap} which has two sub-concepts in two different domains. Using Algorithm 3, the elements of the sub-concepts for each known leaf concept in $\mathcal{C}$ is calculated as follows.

- Leaf concepts '`Ilex`', '`Tilia`', and '`Nerium`' are represented in δ_1 as, respectively: $C_{\mathtt{ia}} = \{c^1_{\mathtt{ia}} :\ \langle \eta^1_{\mathtt{ia}}, \phi^1_{\mathtt{ia}} \rangle\}$, $C_{\mathtt{tt}} = \{c^1_{\mathtt{tt}} :\ \langle \eta^1_{\mathtt{tt}}, \phi^1_{\mathtt{tt}} \rangle\}$, and $C_{\mathtt{no}} = \{c^1_{\mathtt{no}} :\ \langle \eta^1_{\mathtt{no}}, \phi^1_{\mathtt{no}} \rangle\}$.
- Leaf concept '`Acer`' is represented in two domains δ_2 and δ_3 as: $C_{\mathtt{ap}} = \{c^2_{\mathtt{ap}} :\ \langle \eta^2_{\mathtt{ap}}, \phi^2_{\mathtt{ap}} \rangle,\ c^3_{\mathtt{ap}} :\ \langle \eta^3_{\mathtt{ap}}, \phi^3_{\mathtt{ap}} \rangle\}$.
- Leaf concept '`Quercus`' is represented in δ_2 as: $C_{\mathtt{qr}} = \{c^2_{\mathtt{qr}} :\ \langle \eta^2_{\mathtt{qr}}, \phi^2_{\mathtt{qr}} \rangle\}$.
- Leaf concept '`Salix`' is represented in δ_3 as: $C_{\mathtt{sa}} = \{c^3_{\mathtt{sa}} :\ \langle \eta^1_{\mathtt{sa}}, \phi^1_{\mathtt{sa}} \rangle\}$.

In these representations, for example, $\eta^2_{\mathtt{qr}}$ shows the 2D convex polygon of leaf concept '`Quercus`' within δ_2 (see Figure 10). Also, as an example for the weights, $\phi^2_{\mathtt{qr}} = \{\omega_{\delta_2}(C_{\mathtt{qr}}, q_{\mathtt{so}}),\ \omega_{\delta_2}(C_{\mathtt{qr}}, q_{\mathtt{if}})\}$ shows the salience between leaf concept '`Quercus`' and two quality dimensions '`solidity`' and '`isoperimetric factor`' within δ_2. In Figure 10, the graphical presentation of leaf concepts is shown by illustrating the convex hulls of their corresponding sub-concepts.

Now, with the provided elements, the conceptual space of the leaf data set is presented as: $S^{leaf} = \langle\ \mathcal{Q}^l,\ \Delta^l,\ \mathcal{C}^l,\ \Gamma^l\ \rangle$.

5.1.2 Semantic Inference for Unknown Leaf Samples

The inference step aims to derive a semantic description for unknown observations using the developed conceptual space. We present the utility of the conceptual space of leaves using a set of unknown leaf samples from the plant species. Figure 11 presents the selected set of unknown leaf samples to be represented. Here, we show how to apply the inference approach to one of the samples (e.g., leaf (a) in Figure 11). According to Algorithms 4 and 5, the unknown observation (a) is firstly vectorised to an instance γ_a. Then a linguistic description for a is inferred in two phases: setting symbol vectors by inferring in conceptual space and setting the lexical items by inferring in symbol space. Instance γ_a is a set of points within the domains $\Delta^l(\gamma_a)$, denoted: $\gamma_a = \{p^1_{\gamma_a}, p^2_{\gamma_a}, p^3_{\gamma_a}\}$, where the numeric values of each point are the feature values of (a) for each quality dimension in S^{leaf}. For example in δ_2, $p^2_{\gamma_a}{=}\langle q_{\texttt{so}}(a),\ q_{\texttt{if}}(a)\rangle{=}\langle 0.86,\ 0.45\rangle$.

Inference in Conceptual Space of Leaves Here we determine whether γ_a is included in any defined concept's regions, and then infer the semantic labels based on the inclusion of the instance to the regions. Considering the leaf sample (a) in Figure 11, γ_a belongs to the sub-concept $c^1_{\texttt{tt}}$ in δ_1, however, it does not belong to any sub-concept in δ_2 and δ_3. Thus, based on Algorithm 4, the symbol vector for γ_a is set as follows: In the concept layer, using the graded membership function (defined in definition 7): $\mathcal{V}_{\gamma_a,\mathcal{C}}(C_{\texttt{tt}}) = (\text{`}\texttt{Tilia Tomentosa}\text{'}, 0.85)$. In the quality layer, for the quality dimensions of $\delta_2$ and $\delta_3$, using the graded quality function (defined in definition 8): $\mathcal{V}_{\gamma_a,\mathcal{Q}}(q^2_{if}) = (\text{`}\texttt{tipped/toothed}\text{'}, 0.75)$, $\mathcal{V}_{\gamma_a,\mathcal{Q}}(q^2_{so}) = (\text{`}\texttt{smooth edges/entire}\text{'}, 0.86)$, and $\mathcal{V}_{\gamma_a,\mathcal{Q}}(q^3_{lo}) = (\text{`}\texttt{low lobedness}\text{'}, 0.6)$.

Inference in Symbol Space of Leaves By retrieving the information of the symbol vector $\mathcal{V}(\gamma_a)$, we are able to verbalise the elements of symbol vectors into a set of natural language descriptions. As mentioned in Section 4.2.2, we annotate γ_a using the values of $\mathcal{V}_{\gamma_a,\mathcal{C}}$, and characterise it by the values of $\mathcal{V}_{\gamma_a,\mathcal{Q}}$. In particular, the annotation set is $\mathcal{T}_{\mathcal{C}}(\gamma_a) = \{\text{`}\texttt{Tilia Tomentosa}\text{'}\}$, and the characterisation set will be $\mathcal{T}_{\mathcal{Q}}(\gamma_a) = \{\ \text{`}\texttt{tipped}\text{'},\ \text{`}\texttt{smooth edges}\text{'},\ \text{`}\texttt{low lobedness}\text{'}\ \}$. Then the realisation for γ_a is as follows: $\mathcal{T}_{\gamma_a}$ = '`like Tilia Tomentosa(Silver Lime)`, `also with smooth and tipped edges`, `and low lobedness`'.

The same approach is applicable to other unknown leaf samples (shown in Figure 11) to describe them in natural language. Table 1 presents the generated descriptions derived from the semantic inference.

5.2 Physiological Time Series Data Set

In the field of time series data mining, the perception-based analysis of patterns attempts to formalise knowledge and simulate human reasoning (Batyrshin et al., 2007). The perceptions can be represented by linguistic descriptions (i.e. words such as low, increasing, most of the time, etc.) of time series patterns. A time series pattern is a subsequence of a

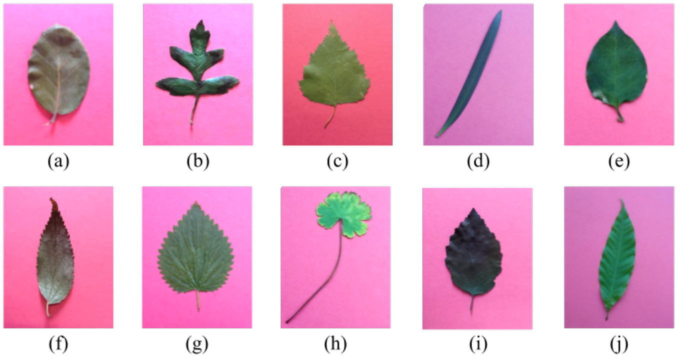


Figure 11: A set of unknown leaf samples.

Leaves	Linguistic Description
Fig. 11(a)	*This leaf is like **Grey Willow**, but it is **round** with a **slightly serrated** margin.*
Fig. 11(b)	*This leaf is like **Japanese Maple**, but it is **oval** with **lobed** margin.*
Fig. 11(c)	*This leaf is like **Silver Lime**, but it is **tipped** with a **slightly toothed** margin.*
Fig. 11(d)	*This leaf is not like any known leaf species, but it is **linear** and **elongated** with **entire** margin.*
Fig. 11(e)	*This leaf is like **Grey Willow**, but it is **round** and **tipped**.*
Fig. 11(f)	*This leaf is not like any known leaf species, but it is **oval** and **tipped** with **toothed** margin.*
Fig. 11(g)	*This leaf is like **Silver Lime**, also it is **tipped** with **low lobed** and **toothed** margin.*
Fig. 11(h)	*This leaf is not like any known leaf species, but it is **round** with **toothed** and **lobed** margin.*
Fig. 11(i)	*This leaf is like **English Holly**, but it is **tipped** with **serrated** margin.*
Fig. 11(j)	*This leaf is like **Oleander**, and it is also **tipped**.*

Table 1: The linguistic descriptions derived for the unknown leaf samples in Figure 11.

univariate time series containing a meaningful behaviour or trend of the data. Many studies consider the problem of qualitative analysis of time series patterns and its manipulation with linguistic information (Yu et al., 2007; Kacprzyk, Wilbik, & Zadrożny, 2008; Banaee & Loutfi, 2015; Batyrshin & Sheremetov, 2008; Novák, 2016). However, in most of them, the required linguistic information is limited by the expert or domain knowledge. In other words, the developed systems are designed to cover specific trends and shapes of patterns with a certain set of the requested linguistic characterisations as the generic vocabulary. The major drawback of such a system is that any other observation (pattern) which is not matched with the provided vocabulary cannot be described and reported in the final summary.

5.2.1 Constructing a Conceptual Space of Time Series Patterns

Assume that we have a set of time series patterns with varying lengths, which are labelled with a set of linguistic terms (See Figure 12). We use a data set of patterns, gathering from physiological sensor data. The time series patterns are exploited from heart rate and respiration rate records in several clinical conditions (Banaee & Loutfi, 2015). This data set includes 78 patterns as known observations with varying time durations, which are categorised in four known classes: *increasing*, *decreasing*, *spike*, and *oscillation*. These class labels of time series patterns are acquired from the common behavioural labels used in the literature of mining and linguistic characterisation of shapes and trends in time series data (Yu, Hunter, Reiter, & Sripada, 2003; Batyrshin et al., 2007; Novák, 2016; Gregory & Shneiderman, 2012). Formally, $\mathcal{D}^p = \{o_1, \ldots, o_{78}\}$ and $\mathcal{Y}^p = \{\ y_{\texttt{in}} : \text{‘}\texttt{Increasing}\text{’},\ y_{\texttt{de}} : \text{‘}\texttt{Decreasing}\text{’},\ y_{\texttt{sp}} : \text{‘}\texttt{Spike}\text{’},\ y_{\texttt{os}} : \text{‘}\texttt{Oscillation}\text{’}\ \}$.

Although there many other types of behaviours for time series patterns, these four classes are chosen in order to simplify the process of conceptualising the patterns. Figure 12 shows the typical examples of such patterns for each of the mentioned classes of the data set. Regarding the set of class labels $\mathcal{Y}^p$, the set of concepts is defined as: $\mathcal{C}^p = \{C_{\texttt{in}},\ C_{\texttt{de}},\ C_{\texttt{sp}},\ C_{\texttt{os}}\}$.

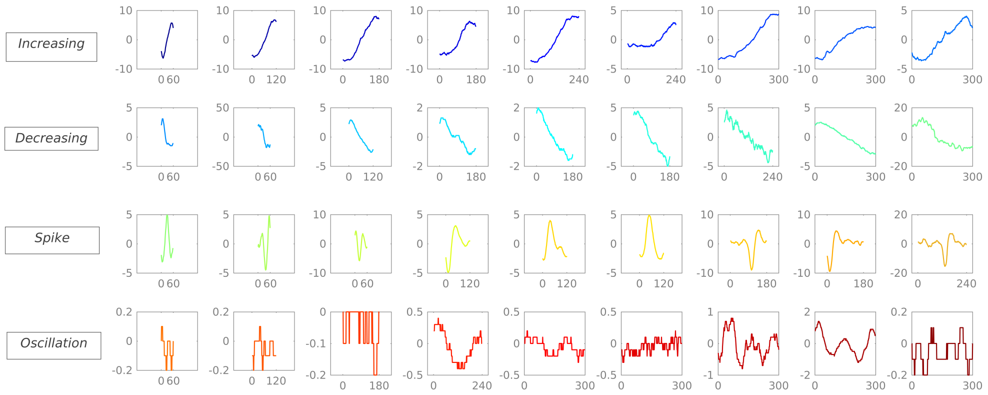


Figure 12: Four set of time series patterns, presenting the known classes of patterns in the data set.

Initialising the primitive set of characteristic features is another input to build the conceptual space of time series patterns. There are many characteristic features for analysing and modelling time series data, from simple statistical features to frequency related ones. As mentioned for the leaf data set, the criterion is how describable or interpretable the features are in linguistic form. For example, in the data set of time series patterns, the values of *integral* or mean features can be useful for analytical tasks in time series mining, but these values are meaningless to the end user of the system in order to visualise or distinguish it from other patterns. In contrast, a feature like the *slope* of a pattern is perceptually interpretable for the user in natural language. Among the various features in the literature of feature-based time series data mining (Fulcher & Jones, 2014; Shumway & Stoffer, 2010; Gregory & Shneiderman, 2012; Nanopoulos, Alcock, & Manolopoulos, 2001), the following features have been chosen as the initial set of features $\mathcal{F}^p = \{X_i = \langle H_{X_i}, I_{X_i}\rangle\}$:

X_α : $\langle$‘$\mathtt{Slope}$’, $(-\pi, \pi)\rangle$ (the slope of the pattern),
$X_{\Delta\mathtt{mm}}$: $\langle$‘$\mathtt{Min-Max\ Diff}$’, $[0, \inf)\rangle$ (the absolute difference between min and max values),
$X_{\Delta\mathtt{se}}$: $\langle$‘$\mathtt{Start-End\ Diff}$’, $[0, \inf)\rangle$ (the difference between start and end values),
$X_{\Delta\mathtt{t}}$: $\langle$‘$\mathtt{Time\ interval}$’, $(0, \inf)\rangle$ (time duration of pattern),
$X_{\mathtt{en}}$: $\langle$‘$\mathtt{Entropy}$’, $[0, \inf)\rangle$ (how chaotic is the pattern),
$X_{\mathtt{fft}}$: $\langle$‘$\mathtt{Frequency}$’, $[0, 1]\rangle$[9],
$X_{\partial x}$: $\langle$‘$\mathtt{First\ Derivative}$’, $[0, 1]\rangle$,
$X_{\partial\partial x}$: $\langle$‘$\mathtt{Second\ Derivative}$’, $[0, 1]\rangle$,
X_σ : $\langle$‘$\mathtt{Standard\ Deviation}$’, $[0, \inf)\rangle$.

Domain Specification for Time Series Pattern Data Set After calculating all these features for every known observation, we have applied the conceptual space construction approach with the inputs of known labelled observations $\mathcal{D}^p$, label set $\mathcal{Y}^p$, and feature set $\mathcal{F}^p$. Figure 13 illustrates the created bipartite graph, which presents the specified domains and quality dimensions. Two selected maximum bicliques determine two domains $\Delta = \{\delta_1, \delta_2\}$, where each domain is specified as follows:

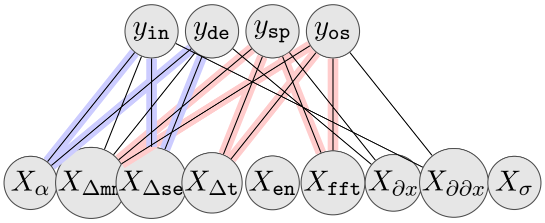


Figure 13: The bipartite graph presenting the relevance of the features and the labels in data set of time series patterns. Also, two chosen bicliques (as the domains) are highlighted with the blue and red edges.

- Domain $\delta_1 = \langle \mathcal{Q}(\delta_1), \mathcal{C}(\delta_1), \omega_{\delta_1} \rangle$, wherein $\mathcal{Q}(\delta_1) = \{q_\alpha,\ q_{\Delta\mathtt{se}}\}$ and $\mathcal{C}(\delta_1) = \{C_{\mathtt{in}}, C_{\mathtt{de}}\}$.
- Domain $\delta_2 = \langle \mathcal{Q}(\delta_2), \mathcal{C}(\delta_2), \omega_{\delta_2} \rangle$, wherein $\mathcal{Q}(\delta_2) = \{q_{\Delta\mathtt{t}}, q_{\Delta\mathtt{mm}}, q_{\mathtt{fft}}\}$ and $\mathcal{C}(\delta_2) = \{C_{\mathtt{sp}}, C_{\mathtt{os}}\}$.

Figure 14 depicts a graphical presentation of the determined domains with the corresponding quality dimensions and concepts for the known time series patterns. As an example, δ_1 is specified by two quality dimensions ‘$\mathtt{start-end\ diff}$’ and ‘$\mathtt{slope}$’, and is associated with two concepts ‘$\mathtt{Increasing}$’ and ‘$\mathtt{Decreasing}$’. An example of the calculated weights in a domain is $\omega_{\delta_1}(C_{\mathtt{in}}, q_\alpha) = 0.62$, which shows the salience of the relation between pattern concept ‘$\mathtt{Increasing}$’ and quality dimension ‘$\mathtt{slope}$’ within δ_1. Similar to the leaf conceptual space, although the process of specifying the domains is data-driven, there may be an interpretation for each determined domain. Here, the interpretation of perceived domains is more sensible. For instance, one can say that δ_1 illustrates the *trend direction* of the known patterns, while δ_2 shows the *shape* of the known patterns (see Figure

[9] For some features, I_i needs to be set manually based on a mapping function from feature's values to one value in an interval. For example, the outputs of the Fourier transform (fft) function of a pattern is mapped to the values in the interval [0.1], likewise for first and second derivatives.

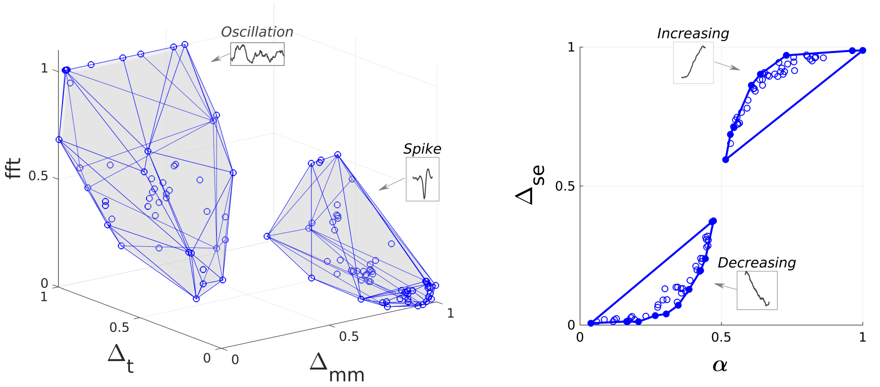


Figure 14: The conceptual space of time series pattern data set: a graphical presentation of the determined domains with the corresponding quality dimensions and concepts.

10). As the output of the domain specification phase for the conceptual space of patterns, the set of quality dimensions will be $\mathcal{Q}^p = \{q_\alpha,\ q_{\Delta\texttt{se}},\ q_{\Delta\texttt{t}},\ q_{\Delta\texttt{mm}},\ q_{\texttt{fft}}\}$. Moreover, the set of instances is defined as: $\Gamma^p = \cup_{y \in \mathcal{Y}}\ \Gamma(y)$, where $|\Gamma^p| = |\mathcal{D}^p|$.

Concept Representation for Pattern Concepts Regarding the output of the domain specification process, each concept in $\mathcal{C}^p$ appears in only one domain (has exactly one sub-concept), as $C_y = \{c_y\}$. By applying Algorithm 3, the elements of the sub-concept for each concept in $\mathcal{C}^p$ is derived as follows.

- Pattern concepts '`Increasing`' and '`Decreasing`' are represented in δ_1 as, respectively: $C_{\texttt{in}} = \{c^1_{\texttt{in}} :\ \langle \eta^1_{\texttt{in}}, \phi^1_{\texttt{in}} \rangle\}$ and $C_{\texttt{de}} = \{c^1_{\texttt{de}} :\ \langle \eta^1_{\texttt{de}}, \phi^1_{\texttt{de}} \rangle\}$.
- Pattern concepts '`Spike`' and '`Oscillation`' are represented in δ_1 as, respectively: $C_{\texttt{sp}} = \{c^2_{\texttt{sp}} :\ \langle \eta^2_{\texttt{sp}}, \phi^2_{\texttt{sp}} \rangle\}$ and $C_{\texttt{os}} = \{c^2_{\texttt{os}} :\ \langle \eta^2_{\texttt{os}}, \phi^2_{\texttt{os}} \rangle\}$.

In these representations, for example, $\eta^2_{\texttt{inc}}$ shows the 3D convex polytope of pattern concept '`Spike`' within δ_2 (see Figure 14). Also, as an example for the weights, $\phi^2_{\texttt{sp}} = \{\omega_{\delta_2}(C_{\texttt{sp}}, q_{\Delta\texttt{t}}),\ \omega_{\delta_2}(C_{\texttt{sp}}, q_{\Delta\texttt{mm}}),\ \omega_{\delta_2}(C_{\texttt{sp}}, q_{\texttt{fft}})\}$ shows the salience between pattern concept '`Spike`' and three quality dimensions '`time interval`', '`min − max diff`', and '`frequency`' within δ_2. In Figure 14, the graphical presentation of time series pattern concepts is shown by illustrating the convex hulls of their corresponding sub-concepts.

Now, with the provided elements, the conceptual space of the time series pattern data set is presented as: $S^{patterns} = \langle\ \mathcal{Q}^p,\ \Delta^p,\ \mathcal{C}^p,\ \Gamma^p\ \rangle$.

5.2.2 Semantic Inference for Unknown Pattern Samples

The aim is to derive a linguistic description for unknown time series patterns. Figure 15 shows a number of examples of unknown patterns that are chosen to be considered. According to Algorithms 4 and 5, an unknown pattern sample (e.g., pattern (a) in Figure

15) is first vectorised to an instance γ_a. Then, a linguistic description for (a) is inferred in two phases: set the values of symbol vector and set the lexical items, by inferring in conceptual space and symbol space, respectively. γ_a is a set of points within $\Delta^p(\gamma_a)$ as: $\gamma_a = \{p^1_{\gamma_a}, p^2_{\gamma_a}\}$, where the numeric values of each point are the feature value of (a) for each quality dimension. For example, in δ_1: $p^2_{\gamma_a}{=}\langle q_\alpha(a),\ q_{\Delta\texttt{se}}(a)\rangle{=}\langle 0.34, 0.28\rangle$.

Inference in Conceptual Space of Patterns Considering the pattern sample (a) in Figure 12, γ_a belongs to the sub-concept $c^1_{\texttt{de}}$ in δ_1, but it does not belong to any sub-concept in δ_2. Based on Algorithm 4, the symbol vector for γ_a is set as follows: In the concept layer, using the graded membership function (defined in Definition 7): $\mathcal{V}_{\gamma_a,\mathcal{C}}(C_{\texttt{de}}) = (\text{`}\texttt{Decreasing}\text{'}, 1)$. In the quality layer, for the quality dimensions of $\delta_2$, using the graded quality function (defined in Definition 8): $\mathcal{V}_{\gamma_a,\mathcal{Q}}(q^2_{\Delta\texttt{t}}) = (\text{`}\texttt{long}\text{'}, 0.75)$, $\mathcal{V}_{\gamma_a,\mathcal{Q}}(q^2_{\Delta\texttt{mm}}) = (\text{`}\texttt{medium range}\text{'}, 0.62)$, and $\mathcal{V}_{\gamma_a,\mathcal{Q}}(q^2_{\texttt{fft}}) = (\text{`}\texttt{fluctuate}\text{'}, 0.7)$.

Inference in Symbol Space of Patterns By retrieving the information of symbol vector $\mathcal{V}(\gamma_a)$, we are able to verbalise the elements of symbol vectors into a set of natural language descriptions. As mentioned in Section 4.2.2, we annotate γ_a using the values of $\mathcal{V}_{\gamma_a,\mathcal{C}}$, and we characterise it by the values of $\mathcal{V}_{\gamma_a,\mathcal{C}}$. In particular, the annotation set is $\mathcal{T}_C(\gamma_a) =$'`Decreasing`', and the characterisation set will be $\mathcal{T}_\mathcal{Q}(\gamma_a) = \{$ '`long`', '`medium range`', '`fluctuate`' $\}$. Then the realisation for γ_a is as follows: $\mathcal{T}_{\gamma_a} =$ '`Decreasing, also fluctuates and it is long with medium range`'.

Table 2 present more results derived from the semantic inference in the conceptual space for the time series patterns shown in Figure 15.

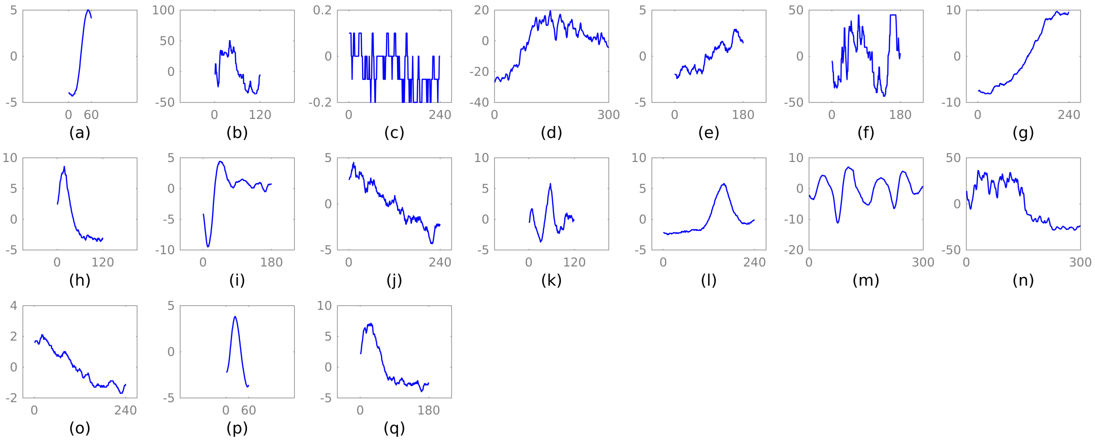


Figure 15: A set of unknown samples of time series patterns.

Patterns	Linguistic Description
Fig. 15(a)	*This pattern is an **Increasing** pattern, but it is **smooth** and very **short**, within a **medium range** of values.*
Fig. 15(b)	*This pattern is like a **Spike** pattern, but it is **noisy** with a **sharp decreasing** trend in a **high range** of values.*
Fig. 15(c)	*This pattern is an **Oscillation** pattern, within a **very short range** of values.*
Fig. 15(d)	*This pattern is an **Increasing** pattern, but it **fluctuates** in a **very long duration**, within a **large range** of values.*
Fig. 15(e)	*This pattern is an **Increasing** pattern, but it **fluctuates** within a **medium range** of values.*
Fig. 15(f)	*This pattern is not like any known pattern, but it **fluctuates** within a **high range** of values.*
Fig. 15(g)	*This pattern is an **Increasing** pattern, but it is **long**, within a **medium range** of values.*
Fig. 15(h)	*This pattern is like **Decreasing Spike** pattern, in a **short duration**.*
Fig. 15(i)	*This pattern is like a **Spike** pattern, but it has a **sharp rise** and **fluctuation**, within a **medium range** of values.*
Fig. 15(j)	*This pattern is a **Decreasing** pattern, but it **fluctuates** in a **long duration**, within a **medium range** of values.*
Fig. 15(k)	*This pattern is a **Spike** pattern, with the **same start and end** values.*
Fig. 15(l)	*This pattern is like a **Spike** pattern, but it has a **smooth** and **slow increasing** trend, within a **medium range** of values.*
Fig. 15(m)	*This pattern is not like any known pattern, but it **wavy** within a **medium range**, and **very long duration**. Also, it has the **same start and end** values.*
Fig. 15(n)	*This pattern is not like any known pattern, but it has a **normal decreasing** trend in a **very long duration**. Also, it is **very fluctuating** in a **large range** of values.*
Fig. 15(o)	*This pattern is like **Decreasing Oscillation** pattern.*
Fig. 15(p)	*This pattern is like a **Spike** pattern, but it is **very short** and **smooth**, within a **high range** of values*
Fig. 15(q)	*This pattern is like **Spike** and **Decreasing** patterns.*

Table 2: The linguistic descriptions derived for the unknown samples of time series patterns in Figure 15.

6. Empirical Evaluation

Assessing the benefits of the proposed conceptual space representation is not a straightforward task. Instead, we evaluated the usefulness of the constructed conceptual spaces for the case studies in Section 5 via the linguistic descriptions derived from such spaces. The experiment evaluates the following aims: (1) to measure the *feasibility* of deriving *accurate* descriptions to *distinguish* unknown observations, and (2) to assess the *goodness* of the descriptions derived from conceptual spaces in comparison to the descriptions derived from other base-line models. To these ends, we conducted a survey, in which participants were asked to

1. identify specific leaf or pattern based on their linguistic description derived from the conceptual space, and
2. rate the goodness of descriptions produced by different models on a Likert scale.

6.1 Survey: Design and Procedure

The survey was conducted online[10]. The participants first had the choice between the leaf and the pattern data set. After an introduction to the used vocabulary, the participant self-evaluated their familiarity with the terminology. The main body of the survey was composed of two parts.

The first part is designed as a set of 4 *multiple choice* questions wherein we asked the participants to read the conceptual space description of a randomly selected sample (among 15 leaves or 17 patterns) and to choose the corresponding image of the leaf or pattern from four shown options. The three incorrect options were also randomly selected from a pool of unknown examples. This task-based (or extrinsic) evaluation (Reiter & Belz, 2009) allowed us not only to measure how well the participants can connect a description of an unknown sample to its correct image but to investigate the incorrectly identified examples and their relation in the conceptual space.

For the second part, we have designed a set of *rating scale* questions, again using four questions per participant. In each question, an image of one unknown observation is randomly selected and shown to the participants, along with the three generated descriptions for that observation from three different models. Participants then are asked to rate each text from 1 to 5 (Likert-scale scoring) regarding how well each of the descriptions helps them to refer to the image. This simultaneous human-rating (or intrinsic) evaluation (Reiter & Belz, 2009) of descriptions enables us to compare the approaches relative to each other, as well as to evaluate the absolute goodness of each approach.

We received 207 responses[11], out of which 102 valid responses have been considered for the *leaf* data set and 98 valid responses for the *pattern* data set. The survey was publicly distributed online to anyone who was interested in participating. The outcome of the demographic questions showed that most of the participants were in the range of 25-44 years old and they are mostly educated in computer science or equivalent. Besides, most of the participants were fluent in English speaking. About the expertise level of the participants, the results show that the participants were more familiar with the terminology that have been used for *pattern* data set, rather than *leaf*. For the *leaf* data, 20% knew none of the lexical items, 70% knew few or some of them, and 10% almost all of them. However, for the *pattern* data, 30% knew few or some of them, 43% knew most of them, and 27% knew all

[10] The survey can be accessed at https://survey.bana.ee

[11] We do not know the exact number of individuals since each participant could decide to perform one of the data sets each time or even redo it with a new set of random questions. Assuming that each person performed both of data sets but just once, the upper bound for the number of participants will be approximately 100 people.

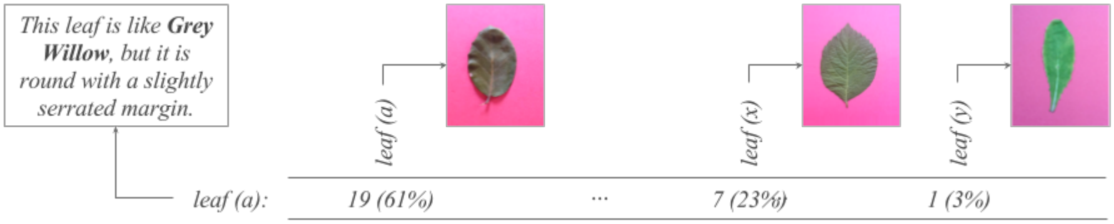

This leaf is like **Grey Willow**, but it is round with a slightly serrated margin.
leaf (a)
leaf (x)
leaf (y)
leaf (a):
19 (61%)
...
7 (23%)
1 (3%)

Figure 16: The description of *leaf (a)* was shown 31 times to the participants. In 19 responses, the correct image of *leaf (a)* is chosen by the participants (61%). The most common misidentified example (7 responses, 23%) was the image of *leaf (x)*, which subjectively is quite similar, and interestingly, it is the closest instance to the *leaf (a)* in the conceptual space. However, the closest instance to *leaf (a)* in the full feature space is *leaf (y)* that its image is rarely misidentified by the participants (only 1 response, 3%).

the introduced terminology. This evidence means that the lexicon used in the descriptions of the patterns was more familiar to the participants.

6.2 Results: Identifying Observations from Linguistic Descriptions

Participants were able to successfully identify all the unknown observations (15 leaves and 17 patterns) with the help of the corresponding conceptual space descriptions. The success rate to identify the correct image for each description in the *leaf* data set was 73% ± 13%, and the success rate in the *pattern* data set was 79% ± 11%. It is reasonable to assume that the higher familiarity of participants with the pattern data set is a possible explanation for the better success rates.

For further investigation of the incorrectly identified (i.e., misidentified) examples, we calculated the geometrical similarity of these answers to the correct one in the conceptual space (multi-domain). According to Gärdenfors (2000), the similarity in conceptual spaces can be calculated by applying Euclidean distance with the domains and applying city-block distance between them. To asses the similarities in conceptual space, we also calculated the geometrical similarity of the same instances but in a full feature space (single-domain) by applying Euclidean distance. We obtained two interesting results: First, the misidentified examples are not uniformly distributed between all possible choices, but instead, participants tended to make similar mistakes. Second, the common misidentified examples are most of the times (73% leaves, 76% patterns) the closest instance to the correct one in the conceptual space. In the full feature space, this was only occasionally true (46% leaves, 29% patterns). This result shows that the confused examples with each other are commonly the nearest instances within the multi-domain conceptual space, which is mostly not the case in the full feature space. One example regarding to this outcome is illustrated in Figure 16.

The results from the first part of the survey show that the proposed conceptual space representation a) is applicable in order to derive semantic descriptions for unknown observations, and b) is suitable in order to represent the cognitively similar observations among the multiple domains.

6.3 Results: Rating Various Linguistic Descriptions for an Observation

In the results of the rating scale questions, we compare the description derived from the conceptual space model with the descriptions derived from the two models of concept formation within the full feature space, one using a generative model and the other a discriminative model (Jebara, 2012; Ng & Jordan, 2016). The *generative* model forms the concepts by modelling the distribution of individual classes (i.e., bound each of them with a convex region). Then, a new observation either belongs to one class or none of them. On the other hand, the *discriminative* model forms the concepts by learning the (hard or soft) boundary between classes (i.e., divide them to Voronoi regions). Hence, a new observation always belongs to at least one class label.

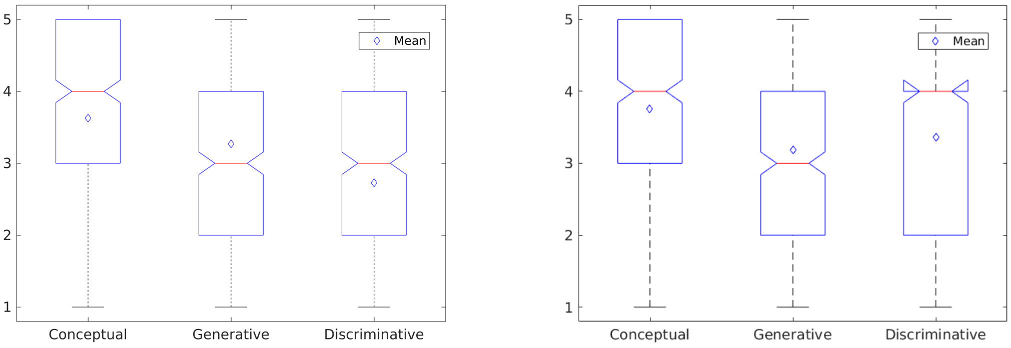
Concerning these two models of concept formation, we have developed two base-line approaches to generate linguistic descriptions. Inspired by the idea of *fuzzy sets* for *linguistic description of data* (Ramos-Soto, Bugarin, Barro, & Taboada, 2015; Ramos-Soto et al., 2016), we have extended the generative and discriminative models to quantify the inclusion of new observations. This extension will allow us to verbalise the numeric output of the models with linguistic descriptions in the inference process. In the generative model, fuzzy sets are employed to quantify the numeric values of the features within the full-feature space. We applied the same semantic inference algorithm (Section 4) on this model to derive such descriptions that most likely involves only the quantified terms of the features. In the discriminative model, with the help of fuzzy sets, the model is extended to multi-label classification (Sorower, 2010), which quantifies the membership of the instances only to their associated concepts. We applied the inference algorithm on this model to derive descriptions that involve only the assigned labels of concepts with quantification on their membership degrees.

Example 18. For leaf (a) in Figure 11, here are the output descriptions from three various approaches:
- Conceptual: "*This leaf is like **Grey Willow**, but it is round with slightly serrated margin.*"
- Generative: "*This leaf is round, wide, connected and entire, with a smooth margin and no indentation.*"
- Discriminative: "*This leaf is similar to **English Holly**, also has some similarity to **Grey Willow**.*"

Table 3 shows the statistical summary of the rating scores received for the descriptions derived from each of the approaches in each data set. Also, these scores are depicted in the form of box plots in Figure 17.

Table 3: The overall scores calculated from the rating responses to the different models in each data set. The numbers show the average scores (and standard deviations) in the range of 1 to 5.

	leaf data set	*pattern* data set
Conceptual	3.62 (1.19)	3.76 (1.07)
Generative	3.27 (1.33)	3.18 (1.29)
Discriminative	2.72 (1.24)	3.36 (1.22)



(a) leaf data set

(b) pattern data set

Figure 17: The box plots of the rating scores received for each of the models deriving descriptions in (a) *leaf* data set and (b) *pattern* data set.

Table 4: Summary of the one-way ANOVA and Wilcoxon tests for the rating scores with respect to the models deriving descriptions.

		leaf data set	pattern data set
ANOVA Test	*Conceptual* vs. *Generative & Discriminative*	$F(2, 1221)$=52.82, p<10^{-21}	$F(2, 1173)$=23.72, p<10^{-13}
Wilcoxon Test	*Conceptual* vs. *Generative*	p<10^{-4}	p<10^{-12}
	Conceptual vs. *Discriminative*	p<10^{-23}	p<10^{-07}
	Generative vs. *Discriminative*	p<10^{-07}	p>0.05 $(*)$

We have applied an ANOVA test to show that the conceptual space description (*Conceptual*) is significantly preferable rated than the two alternatives (*Generative* and *Discriminative*). Here, we used the Likert-scale scores as the dependent variable and the models as the independent variables (groups). We used a post hoc Tukey test to identify the significant difference between the models. The one-way ANOVA test showed a significant effect of the models on the scores. For both data sets, *Conceptual* has the mean significantly different from *Generative* and *Discriminative*, $p < .0001$ (two-tailed). The details of the test have been shown in Table 4.

Moreover, since the ratings are ordinal, we also carried out a non-parametric test (i.e., *Wilcoxon Test*) to identify the significant differences between ratings by comparing each pair of the scores. Table 4 shows the p-values of this method for each pair of models. The output showed that for both data sets, *Conceptual* is significantly different from *Generative* and *Discriminative* ($p < .0001$). Besides the significant difference of the *conceptual* model, an interesting outcome of the tests is the scores of *Generative* and *Discriminative* for two data sets. In the *leaf* data set, participants have given higher scores to *Generative* than *Discriminative* ($p < .0001$). However, in the *pattern* data set, there is no significant difference between these two models ($p > .05$).

Overall, the results from this part of the survey show that the proposed conceptual space representation a) is an appropriate semantic inference model to derive linguistic descriptions for unknown observations, and b) successfully derives descriptions (from multi-domain space) that are naturally preferred by participants, in comparison to the other alternative models (from single-domain space).

7. Discussion

This work has demonstrated how to generate a conceptual model with which it is possible to create semantic interpretations of new observations. Keßler (2006) states that any data-driven approach to generating conceptual spaces cannot be fully automated and require at some point external (symbolic) information, and the presented approach relies to some degree on pre-defined concepts related to the input data set, too. In the following, we discuss a number of issues inherent to our data-driven approach.

7.1 Interpretation of Features

One important issue to address is to determine how interpretable the selected features are for representing the concepts (Section 3). Inherently, the quality dimensions capture the attributes that can cognitively categorise the concepts (Gärdenfors & Williams, 2001). Thus, it makes sense that the feature selection considers the separability between concepts when generating conceptual spaces in a data-driven manner. At the same time, a data-driven approach cannot be completely separated from meaningful semantics. Hence, a further implicit selection criterion has been to select features that can be expressed in natural language, or as stated by Gärdenfors (2000), features which can be given a meaningful perceptual interpretation. The interpretability of feature is indeed context dependent and occurs in different levels of feature abstractness (Spalding & Ross, 2000). As an example, for a given leaf sample, a *large or small area* of a leaf is not informative whereas knowing the *elongation* or *wideness* enables us to depict a more meaningful description.

7.2 Semantics of Domains

Another issue is how to form the domains in a conceptual space without human perception (Section 3.1). While the quality dimensions can be mapped to the feature selection methods, the domains which are formed by grouping the features should be semantically meaningful. In our approach, grouping features is based on how well a subset of the features distinctly represents the various concepts. However, there still exists the problem of verifying the semantic dependency of the quality dimensions within a domain to realise what quality dimensions are *integral* and what are *separable* without necessarily involving background knowledge. While we do not present a solution to this problem, it has been discussed in the literature. For example, Gärdenfors suggests that the verification of deciding whether two quality dimensions are integral or not can be done by empirical testing based on the subject judgements such as the domain experts, and not necessarily using statistical or analytical techniques (Gärdenfors, 2000). It is seemingly difficult, if not impossible, to realise the semantic dependency of the features analytically. For example, by looking at the values of RGB as the dimensions of the colour domain for a set of observations, there is no indication to realise their semantic relations. At the very least in our approach, with relying on the associations between the observations and features, we consider quality dimensions to be integral if they have high relevance to each other that measured by their high relevance to the concepts. Indeed, solving the issue of how to derive a grouping of features for domain specification, can lead to forming a general solution to the problem of determining an evaluation criterion to choose between the competing conceptual spaces, an issue raised by Gärdenfors (2005).

7.3 Quality and Quantity of Known Instances

On concept representation (Section 3.2), convex regions and the salient weights are induced in a data-driven way by considering the observations as the instances of the concepts. Still, these representations suppose that there are sufficient known instances that are representative of the concepts to determine geometric regions and the weights. This assumption is a crucial point that is inherent in all data-driven methods. In the literature of conceptual spaces, the knowledge-driven approaches have used pure thresholds or cutoffs to determine the regions, as illustrated in the definition of the regions for the *mountain* and *hill* concepts described by Adams and Raubal (2009a). However, in such systems like the leaf data set, it is not trivial to initialise in advance the specific geometric boundaries for the categories of the leaf species as the concepts, or the precise salient weights the provided domains. So, in a data-driven approach it is essential to have an adequate set of the known instances of the concepts, and thus an area that is worth to study is to determine on how to dynamically adjust a conceptual space when a set of new observations emerge.

7.4 Concepts as Nouns or Adjectives

One point related to the lexicalisation and the realisation (Section 4) is that from the natural language point of view, the concepts in a conceptual space typically represent the *nouns* while the sub-concepts or properties are related to the *adjectives.* Our approach to determining linguistic descriptions does not make the distinction between adjectives and nouns, and consequently, between the properties and concepts for the class labels from the input learning data set[12]. Rather, one region within a domain is considered as a concept or sub-concept that can be both nouns or adjectives, and be used to describe an observation. For example, descriptions of an object in a conceptual space of *fruits* can be either "*the object is an apple*" or "*the object is red*". In the former, the label refers to the concept *apple* as a noun, and in the latter one, the label refers to the sub-concept *red* as an adjective. We assumed that all the linguistic terms of the quality dimensions are considered as the adjectives. For instance, if *weight* is involved in the description of an object, the term *heavy* is considered as the adjective in such descriptions like "*the object is heavy*".

7.5 Generalisation

While the proposed approach has been tested on two case studies in order to verify its plausibility (Section 5), it would be necessary to identify a general class of problems that

[12] The reason is that from machine learning point of view, there is no linguistic distinction between the different types of the classes, in a sense that the learning problem is to classify whether the *noun* concepts, *adjective* concepts, or even *verb* concepts. For example, in *Iris* data set the classes are nouns, as each class refers to a name of iris plant (Lichman, 2013), however, in *Wine quality* data set the classes are adjectives, as each class refers to the quality level of wines from excellent to poor quality (Cortez, Cerdeira, Almeida, Matos, & Reis, 2009).

the proposed approach can address. Many AI problems dealing with numeric data as the input of learning systems require semantic interpretation for these data, which is needed for interaction with humans. However, in most of the cases, there is no a priori or expert knowledge to explain the aspects of the input observations. This lack of knowledge is more problematic when *connectionist* approaches are applied, since they cannot explain what the learnt emerging model represents. Hence, the introduced framework to construct and utilise conceptual spaces is generally applicable for those AI applications wherein: (1) there is a need for *concept learning* and *concept description* only based on the known available observations, and (2) there is a lack of *interpretability* while creating a learning model, as well as the lack of *explainability* while testing the model by completely unknown observations.

8. Conclusions

This paper presented an automatic approach to map numerical observations into conceptual representations, which is increasingly important in many applications involving human interaction. We have addressed *concept learning* and *semantic interpretation* in AI by deriving conceptual spaces as a semantic model. We have introduced a data-driven *construction* of conceptual spaces from known observations in a given numeric data set. The derived conceptual space is then extended as a semantic *inference* model to generate linguistic descriptions for further unknown observations in natural language.

The framework we proposed employed machine learning algorithms for the task of identifying relevant features and concepts in a numerical data set, in order to shape the domains and quality dimensions of conceptual space. We have argued that the selected and grouped features providing discrimination between concepts are adequate to specify the domains and dimensions while preserving the semantic interpretation of the features and concepts. Then, we have shown an instance-based approach for the task of concept formation within the derived conceptual space. A key finding from the data-driven construction of conceptual space is that it provides a generalisation for concept representation, where the model can be constructed or extend by different types of input instances. With regard to the construction of the conceptual spaces, we have further proposed a semantic inference mechanism to derive linguistic descriptions for new observations. We have shown that linking a symbol space to the conceptual space facilitates the process of inferring linguistic characterisations of the unknown observations. One advantage of such an inference model is that the proposed approach concerns which interpretable aspects of the provided contents can be exploited to describe a phenomenon.

We demonstrated the feasibility of our approach in two case studies of real-world numerical data sets. By performing an empirical evaluation, we have assessed how well linguistic descriptions that are generated based on the derived conceptual space enable human users first to identify the unknown observations. This identification is then followed by a comparison between different semantic models of generating linguistic descriptions to show how well human users prefer the descriptions from the conceptual space model to refer to unknown observations. The evaluation results indicate that a multi-domain representation of concepts (i.e., conceptual spaces) can lead to a better presentation of output descriptions in comparison to a single-domain representation, since the multi-domain spaces preserve the various semantic aspects of the attributes for a concept, while the others combine all the attributes into a single space.

Extending the presented data-driven approach to other cognitive tasks such as concept combination, inductive inference, and property reasoning is a very promising research direction, especially when it comes to data-driven representations in cognitive architectures. Another direction for the future work would be to assess the quality of the automatically derived domains and dimensions. As Gärdenfors (2005) discussed, there is a need to determine the evaluation criteria to choose among competing conceptual spaces. Our framework has the potential to address this need by defining statistical measures to compare the specified domains in a data-driven manner.

References

Adams, B., & Raubal, M. (2009a). Conceptual space markup language (csml): Towards the cognitive semantic web.. In *ICSC*, pp. 253–260.

Adams, B., & Raubal, M. (2009b). A metric conceptual space algebra. In *Spatial information theory*, pp. 51–68. Springer.

Agostaro, F., Augello, A., Pilato, G., Vassallo, G., & Gaglio, S. (2005). A conversational agent based on a conceptual interpretation of a data driven semantic space. In *Congress of the Italian Association for Artificial Intelligence*, pp. 381–392. Springer.

Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine learning*, *6*(1), 37–66.

Aisbett, J., & Gibbon, G. (2001). A general formulation of conceptual spaces as a meso level representation. *Artificial Intelligence*, *133*(1), 189–232.

Aisbett, J., Rickard, J. T., & Gibbon, G. (2015). Conceptual spaces and computing with words. In *Applications of Conceptual Spaces*, pp. 123–139. Springer.

Banaee, H., & Loutfi, A. (2014). Using conceptual spaces to model domain knowledge in data-to-text systems. In *International Natural Language Generation conference (INLG)*, pp. 11–15.

Banaee, H., & Loutfi, A. (2015). Data-driven rule mining and representation of temporal patterns in physiological sensor data. *IEEE journal of biomedical and health informatics*, *19*(5), 1557–1566.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on*, *5*(4), 537–550.

Batyrshin, I., Sheremetov, L., & Herrera-Avelar, R. (2007). Perception based patterns in time series data mining. In *Perception-based Data Mining and Decision Making in Economics and Finance*, pp. 85–118. Springer.

Batyrshin, I. Z., & Sheremetov, L. (2008). Perception-based approach to time series data mining. *Applied Soft Computing*, *8*(3), 1211–1221.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828.

Bollacker, K. D., & Ghosh, J. (1996). Linear feature extractors based on mutual information. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, Vol. 2, pp. 720–724. IEEE.

Brezolin, J. M. L., Fiorini, S. R., de Borba Campos, M., & Bordini, R. H. (2015). Using conceptual spaces for object recognition in multi-agent systems. In *International Conference on Principles and Practice of Multi-Agent Systems*, pp. 697–705. Springer.

Brown, G. (2009). A new perspective for information theoretic feature selection. In *International conference on artificial intelligence and statistics*, pp. 49–56.

Brown, G., Pocock, A., Zhao, M.-J., & Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, *13*(Jan), 27–66.

Chella, A., Frixione, M., & Gaglio, S. (1997). A cognitive architecture for artificial vision. *Artificial Intelligence*, *89*(1-2), 73–111.

Chella, A., Frixione, M., & Gaglio, S. (2003). Anchoring symbols to conceptual spaces: the case of dynamic scenarios. *Robotics and Autonomous Systems*, *43*(2), 175–188.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, *47*(4), 547–553.

Cover, T. M., & Thomas, J. A. (2006). Elements of information theory 2nd edition..

Cubek, R., Ertel, W., & Palm, G. (2015). High-level learning from demonstration with conceptual spaces and subspace clustering. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2592–2597. IEEE.

Decock, L., & Douven, I. (2014). What is graded membership?. *Noûs*, *48*(4), 653–682.

Delgado, M., Ruiz, M. D., Sánchez, D., & Vila, M. A. (2014). Fuzzy quantification: a state of the art. *Fuzzy Sets and Systems*, *242*, 1–30.

Derrac, J., & Schockaert, S. (2015). Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning. *Artificial Intelligence*, *228*, 66–94.

Diaz-Hermida, F., Pereira-Fariña, M., Vidal, J. C., & Ramos-Soto, A. (2016). Characterizing quantifier fuzzification mechanisms: a behavioral guide for practical applications. *arXiv preprint arXiv:1605.03506*.

Duch, W. (2006). Filter methods. In *Feature Extraction*, pp. 89–117. Springer.

Fleuret, F. (2004). Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, *5*(Nov), 1531–1555.

Fulcher, B. D., & Jones, N. S. (2014). Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, *26*(12), 3026–3037.

Gärdenfors, P. (1995). Three levels of inductive inference. *Studies in Logic and the Foundations of Mathematics*, *134*, 427–449.

Gärdenfors, P. (1997). Symbolic, conceptual and subconceptual representations. In *Human and Machine Perception*, pp. 255–270. Springer.

Gärdenfors, P. (2000). *Conceptual spaces: The geometry of thought*. MIT press.

Gärdenfors, P. (2004). Conceptual spaces as a framework for knowledge representation. *Mind and Matter*, *2*(2), 9–27.

Gärdenfors, P. (2005). Induction, conceptual spaces and ai. *The Dynamics of Thought*, 109–124.

Gärdenfors, P. (2014). *The geometry of meaning: Semantics based on conceptual spaces*. MIT Press.

Gärdenfors, P., & Warglien, M. (2012). Using conceptual spaces to model actions and events. *Journal of Semantics*, ffs007.

Gärdenfors, P., & Williams, M.-A. (2001). Reasoning about categories in conceptual spaces. In *IJCAI*, pp. 385–392.

Gatt, A., & Reiter, E. (2009). Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, pp. 90–93. Association for Computational Linguistics.

Gkatzia, D. (2015). *Data-driven approaches to content selection for data-to-text generation*. Ph.D. thesis, Heriot-Watt University.

Gregory, M., & Shneiderman, B. (2012). Shape identification in temporal data sets. In *Expanding the Frontiers of Visual Analytics and Visualization*, pp. 305–321. Springer.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157–1182.

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2008). *Feature extraction: foundations and applications*, Vol. 207. Springer.

Hampton, J. A. (2007). Typicality, graded membership, and vagueness. *Cognitive Science*, *31*(3), 355–384.

Holender, M., Nagi, R., Sudit, M., & Rickard, J. T. (2007). Information fusion using conceptual spaces: Mathematical programming models and methods. In *Information Fusion, 2007 10th International Conference on*, pp. 1–8. IEEE.

Jebara, T. (2012). *Machine learning: discriminative and generative*, Vol. 755. Springer Science & Business Media.

Kacprzyk, J., Wilbik, A., & Zadrożny, S. (2008). Linguistic summarization of time series using a fuzzy quantifier driven aggregation. *Fuzzy Sets and Systems*, *159*(12), 1485–1499.

Kacprzyk, J., & Zadrożny, S. (2010). Computing with words is an implementable paradigm: fuzzy queries, linguistic data summaries, and natural-language generation. *Fuzzy Systems, IEEE Transactions on*, *18*(3), 461–472.

Keogh, E. (2011). Instance-based learning. In *Encyclopedia of Machine Learning*, pp. 549–550. Springer.

Keßler, C. (2006). Conceptual spaces for data descriptions. In *The cognitive approach to modeling environments (CAME), workshop at GIScience*, pp. 29–35.

LeBlanc, K. (2010). Cooperative anchoring: sharing information about objects in multi-robot systems..

Lee, I. (2005). Data mining coupled conceptual spaces for intelligent agents in data-rich environments. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 42–48. Springer.

Lee, I., & Portier, B. (2007). An empirical study of knowledge representation and learning within conceptual spaces for intelligent agents. In *Computer and Information Science, 2007. ICIS 2007. 6th IEEE/ACIS International Conference on*, pp. 463–468. IEEE.

Lewis, M., & Lawry, J. (2016). Hierarchical conceptual spaces for concept combination. *Artificial Intelligence*, *237*, 204–227.

Lichman, M. (2013). UCI machine learning repository..

Lieto, A., Chella, A., & Frixione, M. (2017). Conceptual spaces for cognitive architectures: A lingua franca for different levels of representation. *Biologically Inspired Cognitive Architectures*, *19*, 1–9.

Luger, G. F. (2005). *Artificial intelligence: structures and strategies for complex problem solving*. Pearson education.

Matthiessen, C. (1990). Two approaches to semantic interfaces in text generation. In *Proceedings of the 13th conference on Computational linguistics-Volume 2*, pp. 322–329. Association for Computational Linguistics.

Minsky, M. L. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI magazine*, *12*(2), 34.

Nanopoulos, A., Alcock, R., & Manolopoulos, Y. (2001). Feature-based classification of time-series data. *International Journal of Computer Research*, *10*(3), 49–61.

Ng, A. Y., & Jordan, M. I. (2016). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes..

Novák, V. (2016). Linguistic characterization of time series. *Fuzzy Sets and Systems*, *285*, 52–72.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, *27*(8), 1226–1238.

Quine, W. V. O. (1969). *Ontological relativity and other essays*. No. 1. Columbia University Press.

Ramos-Soto, A., Bugarín, A., & Barro, S. (2016). Fuzzy sets across the natural language generation pipeline. *Progress in artificial intelligence*, *5*(4), 261–276.

Ramos-Soto, A., Bugarin, A. J., Barro, S., & Taboada, J. (2015). Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Transactions on Fuzzy Systems*, *23*(1), 44–57.

Raubal, M. (2004). Formalizing conceptual spaces. In *Formal ontology in information systems, proceedings of the third international conference (FOIS 2004)*, Vol. 114, pp. 153–164.

Reiter, E. (2007). An architecture for data-to-text systems. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pp. 97–104.

Reiter, E., & Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, *35*(4), 529–558.

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(01), 57–87.

Reiter, E., Dale, R., & Feng, Z. (2000). *Building natural language generation systems*, Vol. 33. MIT Press.

Reiter, E., Sripada, S. G., & Robertson, R. (2003). Acquiring correct knowledge for natural language generation. *Journal of Artificial Intelligence Research*, 491–516.

Rickard, J. T. (2006). A concept geometry for conceptual spaces. *Fuzzy optimization and decision making*, *5*(4), 311–329.

Rickard, J. T., Aisbett, J., & Gibbon, G. (2007). Reformulation of the theory of conceptual spaces. *Information Sciences*, *177*(21), 4539–4565.

Schaffernicht, E., Kaltenhaeuser, R., Verma, S. S., & Gross, H.-M. (2010). On estimating mutual information for feature selection. In *International Conference on Artificial Neural Networks*, pp. 362–367. Springer.

Schockaert, S., & Prade, H. (2013). Interpolative and extrapolative reasoning in propositional theories using qualitative knowledge about conceptual spaces. *Artificial Intelligence*, *202*, 86–131.

Schwering, A., & Raubal, M. (2005). Spatial relations for semantic similarity measurement. In *International Conference on Conceptual Modeling*, pp. 259–269. Springer.

Shaham, E., Yu, H., & Li, X.-L. (2016). On finding the maximum edge biclique in a bipartite graph: a subspace clustering approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 315–323. SIAM.

Shepard, R. N., et al. (1987). Toward a universal law of generalization for psychological science. *Science*, *237*(4820), 1317–1323.

Shumway, R. H., & Stoffer, D. S. (2010). *Time series analysis and its applications: with R examples*. Springer Science & Business Media.

Silva, P. F., Marcal, A. R., & da Silva, R. M. A. (2013). Evaluation of features for leaf discrimination. In *Image Analysis and Recognition*, pp. 197–204.

Sorower, M. S. (2010). A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis.*

Spalding, T. L., & Ross, B. H. (2000). Concept learning and feature interpretation. *Memory & cognition*, *28*(3), 439–451.

Sun, R. (1999). Artificial intelligence: Connectionist and symbolic approaches..

Torkkola, K. (2008). Information-theoretic methods. In *Feature Extraction*, pp. 167–185. Springer.

Trivino, G., & Sugeno, M. (2013). Towards linguistic descriptions of phenomena. *International Journal of Approximate Reasoning*, *54*(1), 22–34.

Webb, A. R. (2003). *Statistical pattern recognition*. John Wiley & Sons.

Weisstein, E. W. (2016). Totally ordered set. From MathWorld–A Wolfram Web Resource..

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wu, S. G., Bao, F. S., Xu, E. Y., Wang, Y.-X., Chang, Y.-F., & Xiang, Q.-L. (2007). A leaf recognition algorithm for plant classification using probabilistic neural network. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pp. 11–16. IEEE.

Yager, R. R. (1982). A new approach to the summarization of data. *Information Sciences*, *28*(1), 69–86.

Yu, J., Hunter, J., Reiter, E., & Sripada, S. (2003). Recognising visual patterns to communicate gas turbine time-series data. In *Applications and Innovations in Intelligent Systems X*, pp. 105–118. Springer.

Yu, J., Reiter, E., Hunter, J., & Mellish, C. (2007). Choosing the content of textual summaries of large time-series data sets. *Natural Language Engineering*, *13*(01), 25–49.

Zadeh, L. A. (1996). Fuzzy logic= computing with words. *Fuzzy Systems, IEEE Transactions on*, *4*(2), 103–111.

Zadeh, L. A. (1997). Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. *Fuzzy sets and systems*, *90*(2), 111–127.

Zadeh, L. A. (2000). Section 1-prospects-from computing with numbers to computing with words-from manipulation of measurements to manipulation of perceptions. *Lecture Notes in Computer Science*, *1804*, 3–40.

Zadeh, L. A. (2001). A new direction in ai: Toward a computational theory of perceptions. *AI magazine*, *22*(1), 73.

Zenker, F., & Gärdenfors, P. (2015). Applications of conceptual spaces. *Cited on*, 25.