Soft Comput
https://doi.org/10.1007/s00500-017-2992-x

Semantic distance between vague concepts in a framework of modeling with words

Weifeng Zhang[1,2] · **Hua Hu**[1] · **Haiyang Hu**[1] · **Jinglong Fang**[1]

Abstract Effectively measuring the similarity or dissimilarity of two vague concepts plays a key step in reasoning and computing with vague concepts. In this paper, we define semantic distances between data instances and vague concepts based on modeling vagueness in a framework called label semantics. We also propose two clustering methods based on these sematic distances, which can cluster data instances and vague concepts simultaneously. To evaluate our approach, we conduct several experimental studies on three datasets including Corel images and labels, Reuters-21578, and TDT2. It is illustrated that the proposed distances have the ability to effectively evaluate sematic similarities between data instances and vague concepts.

Keywords Vague concepts · Label semantics · Semantic distance · Clustering

Communicated by V. Loia.

✉ Hua Hu
huhua@hdu.edu.cn

✉ Haiyang Hu
huhaiyang@hdu.edu.cn

Weifeng Zhang
zwf.zhang@hdu.edu.cn

Jinglong Fang
fjl@hdu.edu.cn

[1] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China

[2] Science and Technology on Communication Information Security Control Laboratory, Jiangnan Electronic Communication Institute, Jiaxing, China

1 Introduction

Vague concepts are philosophically important and fundamental to natural language. The meaning transmitted by even basic words is often intrinsically uncertain (Lawry and Tang 2009). For example, one claims that this apple is "sweet." Then how to come up with a definition of "sweet" to cover tastes that are clearly sweet and exclude tastes that are clearly not sweet? It is obvious that the concept of "sweet" is vague. Describing instances using vague concepts is based on the individual's definition of distance or similarity between concepts. Everyday, we need to choose the most appropriate words to depict objects and instances to transmit our opinion. A typical example is that suppose you are asked to describe an apple you have eaten, and several labels are given, such as sweet, red, green, small, medium and big, some of which you may consider as inappropriate for the apple, while some words you are uncertain whether they are appropriate or not. Once you describes the apple using the label small, that means you think the size of the given apple is sufficiently close to your definition of small for an apple.

How to modeling vague concepts using mathematical models has become a research focus in artificial intelligence community for decades. The dominated approach to modeling vague concepts is that of fuzzy logic, which is developed by Zadeh (1965, 1975, 1996). In that approach, a fuzzy set is powerful to represent a vague concept. One of the keys in fuzzy logic is membership function with values ranging from 0 to 1, which reflects the uncertainty. In this paper, vague concepts are modeled with label semantics, which is proposed by Lawry (2006, 2014). It gives an alternative framework to model vague concepts. In contrast to Zadeh's fuzzy logic, the basis of the label semantics intends to quantify uncertainty by measuring the appropriateness of labels used as the description of a given example. It provides a set of appli-

Published online: 17 January 2018

cable calculus using probability distribution of appropriate label sets, and it has been successfully used in many machine learning applications (Qin and Tang 2014; He and Lawry 2014; Turnbull et al. 2016).

By modeling vague concepts with fuzzy sets, several distance measures between fuzzy sets have been proposed in the last decades and applied to solve problems in artificial intelligence. These distance measures can be divided into two groups. The first one works with differences between membership values at particular points, while the other one is based on differences between cuts at particular levels (Pappis and Karacapilidis 1993; Szmidt and Kacprzyk 2000). In recent years, various new distance measures of fuzzy sets have been proposed by improving the classical distances. McCulloch et al. (2013)) proposed a new distance which takes the direction between sets into account. Francisco and his co-workers proposed a fuzzy distance measure and used it to build fuzzy regression model (Francisco et al. 2016). Since vague concepts are modeled as a group of fuzzy sets in fuzzy logic, all the above distance measures can give the similarity of two vague concepts but have no ability to deal with the similarity/dissimilarity between instances and vague concepts.

A principle motivation for this paper is to defined semantic distances between data instances and vague concepts based on modeling vagueness in a framework called label semantics. The semantic distances are defined to measure the similarity of vague concepts expressed by linguistic label expressions and help computers mimic human judgment process to decide the category of an instance or describe an object. Furthermore, two clustering algorithms are proposed in this paper and experimental results demonstrate the reasonableness of the distance definitions and the effectiveness of our proposed algorithms.

We organize the rest of this paper as follows: Sect. 2 gives an overview of modeling vague concepts with label semantics. In Sect. 3, we first give our semantic distance definition based on label semantics and some important properties of the distance are discussed. Then we illustrate the algorithm to learn the key parameters in our distance definition. Then two semantic distance-based clustering algorithms are proposed in Sect. 4. Several experiments are carried out to demonstrate the effectiveness of our defined distances and clustering algorithms in Sect. 5. Finally we give a conclusion about this paper and plan the future work in Sect. 6.

2 Modeling vague concepts with label semantics

With a shared linguistic context, vague concepts are intrinsic to communication between individuals. Our decision to describe an apple as "sweet" is not generally based on any kind of accurate measurement, but on our daily experience of "sweet" taste. Therefore, vagueness is central of the flexibility of our decision. Label semantic proposed by Lawry (2006) aims to incorporate this robustness and flexibility into intelligent system by modeling vagueness in a framework of modeling with words. The foundation of this theory is that it encodes the meaning of linguistic labels according to how they are used by individuals to convey information. Thus, label semantics focus on the decision-making process. In last decade, researchers have proposed several approaches based on label semantics to solve various artificial intelligence tasks. Qin and Lawry (2005) proposed a decision tree learning with label semantics which makes traditional decision tree more robust and it was used to classify weather radar images (Daniel et al. 2007). Label semantics is also used to solve linguistic rule induction (Qin and Lawry 2008). Most recently, label semantics was adopted to build a model of multiagent consensus for vague and uncertain beliefs which significantly improve the robustness of multiagent system (Crosscombe and Lawry 2016).

In the framework of label semantics, agents are given a finite set of labels $\mathrm{LA} = \{\mathrm{label}_1, \mathrm{label}_2, \ldots, \mathrm{label}_n\}$ to convey their description of instances from an underlying universe Ω. For example, if Ω is the set of all possible height of an adult, then LA could consist of the following words: short, medium, tall. All agents agree to use short to describe an adult man whose height is 150 cm. For an adult man of 170 cm, some agent view this man as short, while the others think medium is appropriate. In this sense, we need label expressions to describe these instances. Label expressions are generated by recursively applying logical connecting linguistic labels. In this case, label expressions contain these compound expressions, such as medium $\wedge$ tall, short $\vee$ medium and not short. short $\vee$ medium may be the best one to describe the man of 170 cm. The formal definition of label expressions is as follows.

Definition 1 *(Label expressions)* Label expressions, LE, is defined as follows:

(i) All basic labels are also label expressions: $\mathrm{label}_i \in \mathrm{LE}$ for $i = 1, 2, \ldots, n$.
(ii) If $\rho, \vartheta \in \mathrm{LE}$ then $\neg\rho, \rho \wedge \vartheta, \rho \vee \vartheta \in \mathrm{LE}$.

In order to map label expressions to a set of label sets, λ-mapping function $\lambda(\theta)$ is defined.

Definition 2 (λ-*mapping*) $\lambda : \mathrm{LE} \rightarrow 2^{2^{\mathrm{LA}}}$ is recursively defined as follows: $\forall \rho, \vartheta \in \mathrm{LE}$

(i) $\forall \mathrm{label}_i \in \mathrm{LA}, \lambda(\mathrm{label}_i) = \{F \subseteq \mathrm{LA} : \mathrm{label}_i \subseteq F\}$
(ii) $\lambda(\neg\rho) = \overline{\lambda(\rho)}$.
(iii) $\lambda(\rho \vee \vartheta) = \lambda(\rho) \bigcup \lambda(\vartheta)$.
(iv) $\lambda(\rho \wedge \vartheta) = \lambda(\rho) \bigcap \lambda(\vartheta)$.

The mass function m_x is the appropriateness of sets of labels to describe x.

Definition 3 *(Mass function)* $\forall x \in \Omega$, a mass function on labels is a mapping $2^{\mathrm{LA}} \rightarrow [0,1]$ which satisfy $\sum_{F\subseteq \mathrm{LA}} m_x(F) = 1$.

The agent's subjective belief that label expression $\vartheta \in \mathrm{LE}$ can be applied to describe x is quantified by appropriateness measure denoted by $\mu_\vartheta(x)$ which is defined based on the λ-mapping and mass function.

Definition 4 *(Appropriateness measure)* The appropriateness measure is a function $\mu : \mathrm{LA} \times \Omega \rightarrow [0,1]$ satisfying

$$\forall \vartheta \in \mathrm{LE}, \quad \forall x \in \Omega, \quad \mu_\vartheta(x) = \sum_{F\in\lambda(\vartheta)} m_x(F) \tag{1}$$

The mass function and appropriateness measure imply an important theorem which indicates that there will always be label expressions to definitely describe any instance from a fully covered universe.

Theorem 1 *Suppose the underlying universe Ω is fully covered by a set of labels LA, and label expressions LE are obtained by recursively applying standard logical connection. Then,*

$$\forall x, \exists \vartheta \in LE, \quad \mu_\vartheta(x) = 1 \tag{2}$$

This theorem can be proved as follows.

Proof (Theorem 1) According to definition of mass function on labels, we can obtain: $\forall x, \exists \mathbf{B} = \{B_1, \ldots, Bn\}, B_i \subseteq \mathrm{LA}, \sum_{i=1}^{n} m_x(B_i) = 1$

Let $\lambda(\vartheta) = \mathbf{B}$, then, $\mu_\vartheta(x) = \sum_{B\subseteq\lambda(\vartheta)} m_x(B) = 1$. □

Consider the example of describing the height of an adult as shown in Fig. 1. In this example, LA = {short, medium, tall}. Now ten people are asked to describe the height of a man of 174 cm. Suppose 8 people think that medium is appropriate to describe this man, while 2 people think that the appropriate labels for this man can be any one of short and medium. According to Definition 3, the mass function for this man is:

$$m_{174} = \{\text{medium}\} : 0.8, \{\text{short, medium}\} : 0.2$$

By using the formula in the definition of appropriateness measure (Def. 4), the appropriateness of applying *medium* to describe a man with 174 cm is

$$\mu_{\text{medium}}(174) = 0.8 + 0.2 = 1$$

and that of *short* is

$$\mu_{\text{short}}(174) = 0.2$$

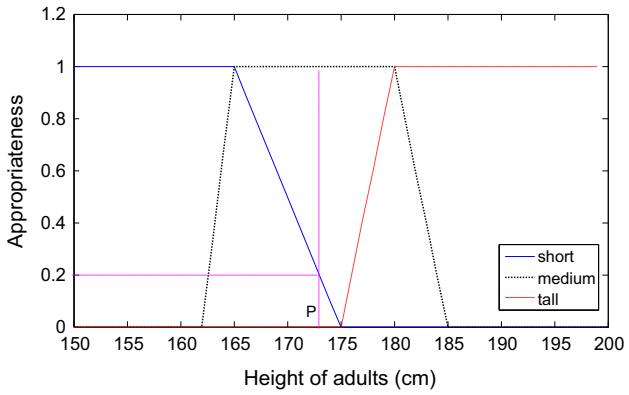


Fig. 1 A continuous universe covered by three trapezoidal labels: *short*, *medium* and *tall*. Given a data point $P = 174$ cm, $m_P = \{\text{medium}\}:0.8\{\text{short, medium}\} : 0.2$. Based on the definition of mass function and appropriateness measure, $\mu_{\text{medium}}(P) = 1$, $\mu_{\text{short}}(P) = 0.2$

It is obvious that $\mu_{\text{label}_i}(x)$ can be uniquely determined, given the mass assignment m_x. But given appropriateness values of all basic labels, we will get an infinite set of mass functions. Hence, a consonant selection function is defined as follows.

Definition 5 *(Consonant selection function)* Given appropriateness measures of basic labels $\mu_{\text{label}_i}(x) : i = 1, \ldots, n$ and satisfy $\mu_{\text{label}_i}(x) \geq \mu_{\text{label}_{i+1}}(x)$, then the mass function can be identified by consonant selection,

(i) $m_x(\{\text{label}_1, \ldots, \text{label}_n\}) = \mu_{\text{label}_n}(x)$
(ii) $m_x(\{\text{label}_1, \ldots, \text{label}_i\}) = \mu_{\text{label}_i}(x) - \mu_{\text{label}_{i+1}}(x)$ for $i = 1, \ldots, n$
(iii) $m_x(\emptyset) = 1 - \mu_{\text{label}_1}(x)$

The basic idea of consonant selection function is that agents first sorts all of the labels by their appropriateness measures, for each $x \in \Omega$. They then choose the labels with highest belief values m_x to describe x. Using this selection function, the following theorem[1] has been proved in the literatures (Lawry 2006).

Theorem 2 $\forall \rho, \vartheta \in LE, \forall x \in \Omega$

(i) $\mu_{\rho\wedge\vartheta}(x) = min(\mu_\rho(x), \mu_\vartheta(x))$
(ii) $\mu_{\rho\vee\vartheta}(x) = max(\mu_\rho(x), \mu_\vartheta(x))$
(iii) $\mu_{\neg\rho}(x) = 1 - \mu_\rho(x)$

In recent years, Lawry and his co-worker proposed a new interpretation of label semantics which incorporates prototype theory (Lawry and Tang 2009; Lewis and Lawry 2016).

[1] Actually the third equation of this theorem doesn't need the selection function assumption. It is a general property of appropriateness measure.

The connotation of prototype theory (Rosch 1973, 1975) is that concepts are represented by prototypes and instances can be classified by measuring their distances to these prototypes. In this approach, distance function is needed to evaluate the semantic divergence between data instances and concepts, and typical exemplars of a concept should have sufficient short distance to the concept's prototypes. In the interpretation of label semantics using prototype theory, each label expression is a natural class and has a set of prototypes. A label expression ϑ is then deemed to be an appropriate description of an instance $x \in \Omega$ if x is sufficiently similar to the prototypes for ϑ. The required of being sufficiently similar is clearly imprecise and is modeled here by introducing an uncertain threshold on distance from prototypes.

In this interpretation framework, agents use a set of label expressions $\mathbb{LE} = \{\text{LE}_1, \text{LE}_2, \ldots, \text{LE}_n\}$ to describe an underlying universe Ω. Each label expression is associated with a set of prototype values $P_i \subseteq \Omega$, and with an uncertain threshold ε_i, which is drawn from probability distribution δ_i. The intuition here is that ε_i captures the idea of being sufficiently similar to prototypes. Given an element $x \in \Omega$, then the appropriateness of using LE_i as the description for this element can be calculated using the following equation, which has been proved in Lawry and Tang's work (Lawry and Tang 2009).

$$\mu_{\text{LE}_i}(x) = \int_{d(x,P_i)}^{\infty} \delta_i(\varepsilon_i) d\varepsilon_i \tag{3}$$

Here $d(x, P_i)$ denotes the distance between element x and prototype P_i. Figure 2 shows this property. Furthermore, the distance function naturally ranks the appropriateness of labels for any instance x, according to which label L_j is more appropriate to describe x than label L_i if x is closer to P_j than to P_i, i.e., $L_j \succ_x L_i$ iff $d(x, P_j) < d(x, P_i)$. $\succ_x$ denotes the appropriateness ordering for instance x. We can define a set of labels whose distances to x do not exceed the threshold ϵ, formally denoted as $D_x^{\epsilon} = \{L_i \in \text{LA} : d(x, P_i) \leq \epsilon\}$. This D_x^{ϵ} can be considered as a random set from $[0, \infty)$ into 2^{LA}. This provides a link to the random set interpretation of fuzzy sets. The difference is that random set maps to sets of labels in this case, rather than sets of elements. Thus, distance function plays an important role in this interpretation of label semantics and also is the foundation of reasoning and computing with label semantics. However, Lawry and Tang (2009) have not given the definition of this distance function. This inspires us to build a series of distance functions to measure semantic divergence between data instances and vague concepts in the framework of label semantics and prototype theory.

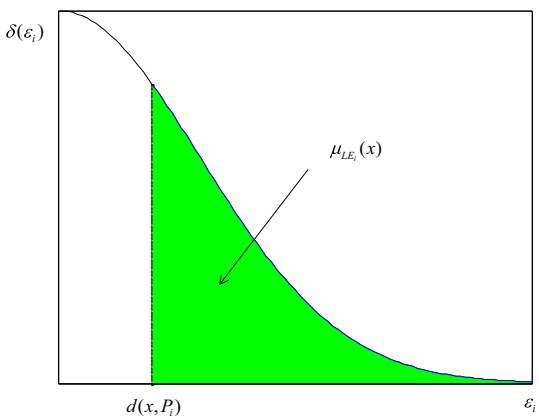


Fig. 2 In the interpretation of label semantics using prototype theory, the appropriateness measure can be calculated by measuring the area under the density function δ

3 Semantic distance between vague concepts

In last decades, many distance or similarity measures (Hyung and Song 1994; Szmidt and Kacprzyk 2000; Li 2004; Janis and Montes 2007; McCulloch et al. 2013; Francisco et al. 2016) have been proposed based on the theory of fuzzy sets. Since vague concepts are modeled as a group of fuzzy sets in fuzzy logic in these existing works, the distance measures mentioned above can give the similarity of two vague concepts but have no ability to deal with the similarity/dissimilarity between instances and vague concepts. In this section, we present a series of measures to evaluate the distance between data instances and vague concepts represented by label expressions.

3.1 Semantic distance

In label semantics, an intelligent agent must go through a decision-making process to choose labels or expressions to describe an object. This procedure is mainly depends on appropriate measures. For this reason, we first define the distance between instances from the underlying universe Ω using the Minkowski distance which has been widely used in fuzzy clustering algorithms (Rosmalen 2006; Groenen and Rosmalen 2007; Srivastava and Pathak 2011).

Definition 6 *(Distance between data instances)* Given two data instances x and y from an universe Ω covered by label set $\mathbb{L} = \{L_1, L_2, \ldots, L_n\}$, then the distance between x and y is defined by:

$$d(x, y) = \left(\sum_{i=1}^{n} |\mu_{L_i}(x) - \mu_{L_i}(y)|^p \right)^{\frac{1}{p}}, \quad p \geq 1, \quad p \neq \infty \tag{4}$$

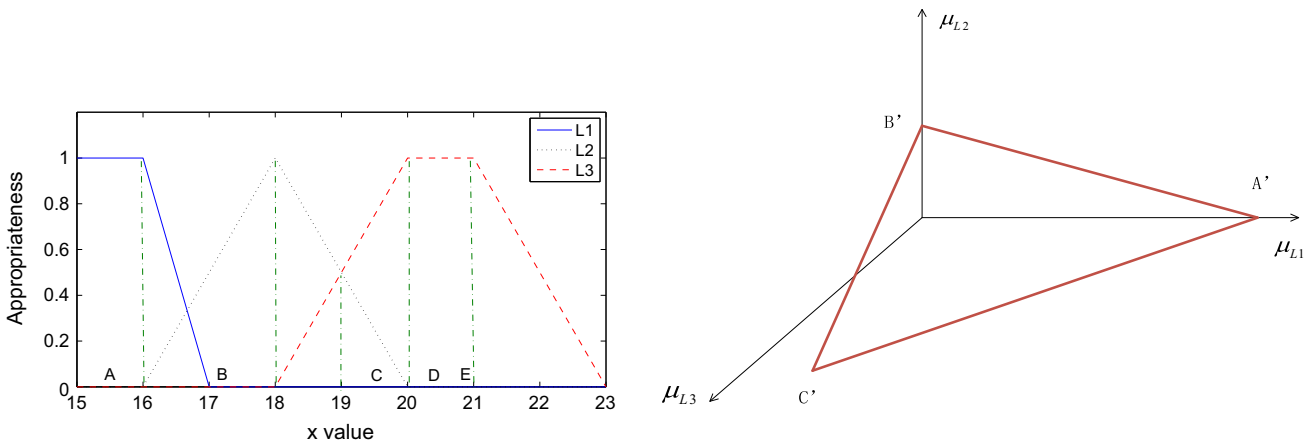


Fig. 3 (Left) An universal fully covered by three labels. Data instances A, B, C, D, E are plotted. (Right) The appropriateness space generated by appropriateness measure. A', B', C' are mapped from A, B, C. Based on our definition of distance between data instances, $d(A, B) = ED(A', B')$, $d(B, C) = ED(B', C')$, $d(A, C) = ED(A', C')$, and $d(D, E) = 0$, when $p = 2$

In Definition 6, the semantic distance between data instances is considered as the L-p norm of difference of these data instances' appropriateness measures. This definition can be considered as two steps. When we are asked to give the distance between two data instances x and y, we first map these data instances from the underlying universe Ω to the appropriateness space by appropriateness measure function $\mu(\cdot)$; thus, x and y can be, respectively, represented as n-dimensional vectors: $< \mu_{L_1}(x), \mu_{L_2}(x), \ldots, \mu_{L_n}(x) >$ and $< \mu_{L_1}(y), \mu_{L_2}(y), \ldots, \mu_{L_n}(y) >$. We then calculate the L-p norm of the difference between these two vectors. Specially this distance will become Manhattan distance when $p = 1$ and become Euclidean distance when $p = 2$. Figure 3 shows an example in which p equals 2, and data instances $A(x = 15.5)$, $B(x = 17.1)$ and $C(x = 19.5)$ on the underlying universal Ω are mapped into appropriateness space and become $A' =< 1, 0, 0 >$, $B' =< 0, 1, 0 >$ and $C' =< 0, 0.2, 0.8 >$. Then the Euclidean distances between A', B', C' are deemed to be the semantic distances between A, B, C. Here we restrict that $p \neq \infty$. L-∞ norm, also called Chebyshev distance, is equivalent to considering the maximum of the data instances' appropriateness measures as their sematic distance which is unexplainable for some data instances. For example, $d(175, 180) = d(175, 190) = 1$ in Fig. 1 if $p = \infty$. In other words, Chebyshev distance cannot give the distinction between the semantics of these three data instances. In Sect. 5, we will evaluate the impact of p on the performance of our proposed clustering algorithms.

Based on Definition 6, we can easily obtain the following properties.

Lemma 1 *$\forall x, y, z \in \Omega$, the distance $d(\cdot, \cdot)$ defined by Definition 6 has the following properties.*

(i) $d(x, x) = 0$
(ii) $d(x, y) \geq 0$
(iii) $d(x, y) = d(y, x)$
(iv) $d(x, y) + d(y, z) \geq d(x, z)$

Here we should pay attention to that the distance defined here is a pseudo-distance because it does not satisfy $d(x, y) = 0$ if and only if $x = y$. For example, the semantic distance between data instances $D(x = 20.1)$ and $E(x = 20.9)$ in Fig. 3 equals zero but $D \neq E$. Actually, two instances' semantic distance equaling zero means they can be definitely described using the same subsets of labels with identical appropriateness, and do not means they are the same one.

Definition 6 gives the way of calculating distance between two data instances in a label space. Now we consider how to extend the measure to evaluate the distance between data instance and vague concept which is represented by label expressions. Based on prototype theory, each label expression has an associated set of prototypes who are the most typical exemplars of the vague concept. A label expression ϑ is then considered to be appropriate to describe an instance $x \in \Omega$ if x is sufficiently similar to the prototypes for ϑ. Thus, the semantic distance between data instance and vague concept can be evaluated using the distances between this data instance and the prototypes of the concept. The following gives the formal definition of distance between data instance and label expression.

Definition 7 *(Distance between data instance and label expression)* Given an data instance x from universe Ω which is covered by label set $\mathbb{L} = \{L_1, L_2, \ldots, L_n\}$ and a label expression ϑ which is recursively generated from labels in $\mathbb{L}$ using standard logical connection, then the distance between x and ϑ is defined by:

$$d(x, \vartheta) = \min\{d(x, y), y \in P_\vartheta\} \tag{5}$$

where $P_\vartheta = \{z, \forall w \in \Omega, \mu_\vartheta(z) \geq \mu_\vartheta(w)\}$.

Here the set of prototypes for a label expression ϑ, denoted as P_ϑ, is further defined as the set of elements who get the highest appropriateness when described using label expression ϑ. As shown in the left part of Fig. 3, the prototypes of L_1 are all the elements in the interval [0, 16], while the prototype of label L_2 is just a point $x = 18$. Furthermore, the prototype of label expression $L_2 \wedge L_3$ is the point $x = 19$. Rather than using average or medium operation, Definition 7 is given based on a minimum operation to guarantee that distances between vague concept and its prototypes equal zero. This property means that data instance x can be definitely described using label expression ϑ if x is a prototype of ϑ or the distance between x and ϑ is zero. For instance, in Fig. 3, data instance A is a prototype of L_1 according to our definition of prototypes. Then the distance between them equals zero based on Definition 7.

Finally, we evaluated the semantic distance between two label expressions by integrating the distances between prototypes of these two expressions.

Definition 8 *(Distance between label expressions)* Suppose a universe Ω is covered by label set $\mathbb{L} = \{L_1, L_2, \ldots, L_n\}$ and a label expression set $\mathbb{LE}$ is recursively generated from labels in $\mathbb{L}$ using standard logical connection. Then the semantic distance between two label expressions ρ and ϑ is defined by:

$$d(\rho, \vartheta) = \frac{d_\rho}{2|P_\rho|} + \frac{d_\vartheta}{2|P_\vartheta|} \tag{6}$$

where

$$d_\rho = \begin{cases} \Sigma_{x \in P_\rho} d(x, \vartheta), & P_\rho \text{ is discrete} \\ \int_{P_\rho} d(x, \vartheta) \mathrm{d}x, & \text{otherwise} \end{cases}$$

$$d_\vartheta = \begin{cases} \Sigma_{x \in P_\vartheta} d(x, \rho), & P_\vartheta \text{ is discrete} \\ \int_{P_\vartheta} d(x, \rho) \mathrm{d}x, & \text{otherwise} \end{cases}$$

$|P_\rho|$ and $|P_\vartheta|$, respectively, represent the number of prototypes of ρ and ϑ. This definition shows that distance between label expressions satisfies nonnegativity, symmetry, but not comply with triangle inequality anymore, which is not strictly required (Lawry and Tang 2009).

So far, the vague concepts discussed in this paper have only one property which can be covered by a set of labels. Actually, vague concepts are usually multidimensional. We frequently describe instance and concept from multiple perspectives in our daily usage of natural language. For example, we may use "sweet" and "red" to describe an apple. That means we use "sweet" to describe the taste of the apple and use "red" to describe its color. To evaluate the distance between these kind of multidimensional vague concept, we first define the multidimensional label expressions to model vague concepts and then give the definition of distance between multidimensional label expressions.

Definition 9 *(Multidimensional label expressions)* $\mathrm{MLE}^{(n)}$ is the set of all multidimensional label expressions that can be generated as follows:

(i) If $\rho \in \mathbb{LE}_i$ for $i = 1, \ldots, n$, then $\rho \in \mathrm{MLE}^{(n)}$.

(ii) If $\rho, \vartheta \in \mathrm{MLE}^{(n)}$, then $\neg\rho, \rho \wedge \vartheta, \rho \vee \vartheta \in \mathrm{MLE}^{(n)}$.

Here we use $\mathbb{LE}_i$ to denote the set of logical expressions used to describe the ith-dimensional property of the vague concept. In the prior example, we can use $\mathrm{MLE}^{(2)} =$ sweet $\wedge$ red as the appropriate description of the apple based on this definition. Now we give the semantic distance between multidimensional data instances, distance between multidimensional instance and label expression, and distance between multidimensional label expressions as follows.

Definition 10 *(Distance between multidimensional data instances)* Given two multidimensional instances $\mathbf{x} =< x_1, x_2, \ldots, x_n >$, and $\mathbf{y} =< y_1, y_2, \ldots, y_n >$, then the distance between $\mathbf{x}$ and $\mathbf{y}$ can be calculated using the following formula:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{D^T \mathrm{AD}} \tag{7}$$

where $D =< d(x_1, y_1), d(x_2, y_2), \ldots, d(x_n, y_n) >^T$ and $A \succeq 0$

Here x_i and y_i represent the value of ith-dimensional property of $\mathbf{x}$ and $\mathbf{y}$, respectively. $d(x_i, y_i)$ means the semantic distance between the ith-dimensional property of the data instance which can be calculated using Definition 6. Then the distance between multidimensional data instances can be considered as a combination of these n distances, using matrix A obtained by metric learning algorithm introduced in Sect. 3.2.2. Restricting A to be positive semi-definite, $A \succeq 0$, is to guarantee this distance a pseudo-metric. Furthermore, A parameterizes a family of Mahalanobis distances. Specially, when we set $A = I$, then the distance is just the classical Euclidean distance. If we restrict A to be diagonal, then we are to learn a metric which gives different "weights" to different properties of vague concepts.

Definition 11 *(Distance between multidimensional element and label expression)* Given a multidimensional instance $\mathbf{x} =< x_1, x_2, \ldots, x_n >$, and a multidimensional label expression $\Phi = \mathrm{LE}_1 \wedge \mathrm{LE}_2 \wedge \cdots \wedge \mathrm{LE}_n \in \mathrm{MLE}^{(n)}$, then

the distance between $\mathbf{x}$ and Φ can be calculated using the following formula:

$$d(\mathbf{x}, \Phi) = \sqrt{D^T \mathrm{AD}} \tag{8}$$

where $D =< d(x_1, \mathrm{LE}_1), d(x_2, \mathrm{LE}_2), \ldots, d(x_n, \mathrm{LE}_n) >^T$ and $A \succeq 0$

Definition 12 *(Distance between multidimensional label expressions)* Given two multidimensional label expressions $\Phi = \mathrm{LE}_1^\Phi \wedge \mathrm{LE}_2^\Phi \wedge \cdots \wedge \mathrm{LE}_n^\Phi > \in MLE^{(n)}$ and $\Upsilon = \mathrm{LE}_1^\Upsilon \wedge \mathrm{LE}_2^\Upsilon \wedge \cdots \wedge \mathrm{LE}_n^\Upsilon \in MLE^{(n)}$, then the distance between Φ and Υ can be calculated using the following formula:

$$d(\Phi, \Upsilon) = \sqrt{D^T \mathrm{AD}} \tag{9}$$

where $D =< d(\mathrm{LE}_1^\Phi, \mathrm{LE}_1^\Upsilon), d(\mathrm{LE}_2^\Phi, \mathrm{LE}_2^\Upsilon), \ldots, d(\mathrm{LE}_n^\Phi, \mathrm{LE}_n^\Upsilon) >^T$ and $A \succeq 0$

The matrix A in Definitions 11 and 12 is same as that in Definition 10. The distance between multidimensional data instance and label expression, and distance between multidimensional label expressions differ from distance between multidimensional data instances only in the definition of D. $d(x_i, \mathrm{LE}_i)$ and $d(\mathrm{LE}_i^\Phi, \mathrm{LE}_i^\Upsilon)$ can, respectively, be calculated using Definitions 7 and 8.

3.2 Parameter learning for semantic distance

Section 3.1 gives our semantic distance definitions for instances, label expressions and multidimensional label expressions. All of these definitions are based on the appropriateness measure. In addition, the matrix A in Definition 10 is yet unknown. In this section, we will give the methods to learn the appropriateness measure function and A from data.

3.2.1 Appropriateness measure function learning

Suppose we have a set of instances $\{x_i\}_{i=1}^m$ which have only one property and are acquired by sampling from the underlying universal Ω which is covered by a label set $\mathbb{L} = \{L_1, L_2, \ldots, L_l\}$. The appropriateness measures of all basic labels for instances are also given as $\widehat{\boldsymbol{\mu}}(x_i) =< \hat{\mu_{L_1}}(x_i), \hat{\mu_{L_2}}(x_i), \ldots, \hat{\mu_{L_l}}(x_i) >, i = 1, 2, \ldots, m$. Thus, the appropriateness measure function learning can be considered as a regression problem which can be solved using various algorithms (David 2005; Bishop 2006; Scott 2012). In this study, we using the support vector regression (SVR) proposed by Vapnik (1998), Cortes and Vapnik (1995), Druker (19997), Smola and Scholkopf (2004), Gu et al. (2015), Gu et al. (2015), Gu and Sheng (2016), Gu et al. (2016) mainly for two reasons: (1) The appropriateness measure functions for different real-world applications are always different and nonlinear. Typically, they can be bell-shaped functions, Gaussian functions, trapezoid functions or any other kinds of functions. SVR can fit these nonlinear functions effectively by kernels which transform the data space to make it possible to perform the linear separation (we use RBF kernel in this paper). (2) SVR performs well on small training data set. When we learn the appropriateness functions, the training data set is only a small part of the given data. For each basic label $L \in \mathbb{L}$, appropriateness measure function learning is to learn $\mu_L(x)$ from the training data set

$$D = \{(x_1, \hat{\mu_L}(x_1)), (x_2, \hat{\mu_L}(x_2)), \ldots, (x_m, \hat{\mu_L}(x_m))\} \tag{10}$$

The function $\boldsymbol{\mu}_L(x)$ can take the form

$$\mu_L(x) = \boldsymbol{\omega}^T \phi(x) + b \tag{11}$$

The $\phi(x)$ denotes a nonlinear transformation used to map the x onto a high-dimensional space and b is the "bias" term. In practice, SVR is formulated as follows.

$$\begin{aligned} \min_{\boldsymbol{\omega}, b, \xi_i, \xi_i^*} \quad & \frac{1}{2} \parallel \boldsymbol{\omega} \parallel + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \\ \text{s.t.} \quad & \mu_L(x_i) - \hat{\mu_L}(x_i) \le \epsilon + \xi_i, \\ & \hat{\mu_L}(x_i) - \mu_L(x_i) \le \epsilon + \xi_i^*, \\ & \xi_i \ge 0, \xi_i^* \ge 0, i = 1, 2, \ldots, m \end{aligned} \tag{12}$$

Similar to support vector machine, the computational complexity is high because of the high dimensionality of $\boldsymbol{\omega}$. So the SVR is usually converted into its dual problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \boldsymbol{\alpha}^*} \quad & \sum_{i=1}^{m} \hat{\mu_L}(x_i)(\alpha_i^* - \alpha_i) - \epsilon(\alpha_i^* + \alpha_i) \\ & - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \kappa(x_i, x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) = 0, \quad 0 \le \alpha_i^*, \alpha_i \le C \end{aligned} \tag{13}$$

where $\kappa(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is the kernel function. The final solution of appropriateness is given by the following equation which limits the appropriateness to the range [0, 1].

$$\mu_L(x) = \max(0, \min(\sum_{i=1}^{m} (\alpha_i^* - \alpha_i) \kappa(x, x_i) + b, 1)) \tag{14}$$

Our data-based learning method for appropriateness measure function is similar to the estimation of membership function for a fuzzy set which is an important step in many applications of fuzzy theory. Most of these estimation meth-

ods (Medasani and JK, Krishnapuram R, 1998; Nieradka and Butkiewicz 2007; Zheng et al. 2015; Guo and XW, Wang L, 2016) discard the dependencies between fuzzy sets. Similarly, we learn appropriateness measure function for each label expression separately regardless of the dependencies between label expressions. It would be interesting to investigate the dependencies using graphical models such as Bayesian networks. However, it is beyond the scope of this paper and probably remains to be studied as a part of our future work.

3.2.2 Learning distance metric for multidimensional label expressions

As shown in Eq. 9, the matrix A is the parameter which needs to be learned from data using metric learning methods (Roweis and Saul 2000; Xing et al. 2002; Weinberger and Saul 2009; Goldberger et al. 2005; Zhang et al. 2012). In this paper, inspired by the algorithm proposed by Xing et al. (2002), we learn the matrix A as follows. We define a simple criterion for the desired metric: make the pairs of multidimensional data instances with similar semantic meaning have small distance between them, while keeping the pairs of data instances with dissimilar semantic meaning apart. This gives the optimization problem:

$$\begin{aligned} \min_{A} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} d^2(\mathbf{x}_i,\mathbf{x}_j) \\ \text{s.t.} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} d(\mathbf{x}_i,\mathbf{x}_j) \geq 1 \\ & A \succeq 0 \end{aligned} \tag{15}$$

where $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ means $\mathbf{x}_i$ and $\mathbf{x}_j$ area semantically similar, while $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{D}$ denotes that the semantic meanings of $\mathbf{x}_i$ and $\mathbf{x}_j$ are dissimilar. The constraint $\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} d(\mathbf{x}_i, \mathbf{x}_j) \geq 1$ is to ensure that A does not collapse the dataset into a single point. $d(\mathbf{x}_i, \mathbf{x}_j)$ follows Definition 10.

In that case, we set $A = \text{diag}(A_{11}, A_{22}, \ldots, A_{nn})$ where A_{ii} denotes the weight of the ith property of $\mathbf{x}_i$ and $\mathbf{x}_j$. The optimization problem can be solved using the Newton–Raphson method (Victor and Semyon 2006). Define

$$\begin{aligned} g(A) &= g(A_{11}, A_{22}, \ldots, A_{nn}) \\ &= \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} d^2(\mathbf{x}_i,\mathbf{x}_j) - \log\Big(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} d(\mathbf{x}_i,\mathbf{x}_j)\Big) \end{aligned} \tag{16}$$

When we want to learn a full matrix A, the Newton's method becomes prohibitively expensive. In this case, we use projected gradient descent to solve the equivalent optimization problem:

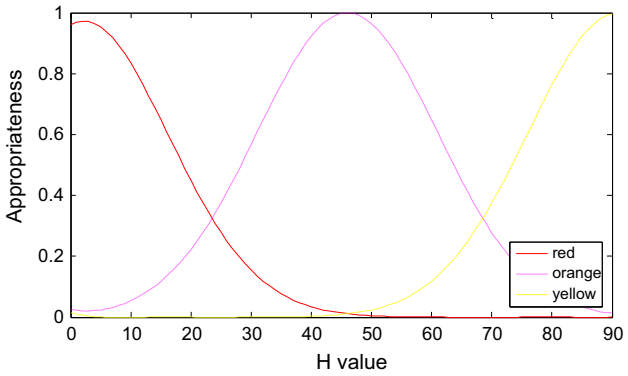


Fig. 4 Predefined appropriateness measures on three basic labels: red, orange, yellow (color figure online)

$$\begin{aligned} \max_{A} \quad & g(A) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} d(\mathbf{x}_i,\mathbf{x}_j) \\ \text{s.t.} \quad & f(A) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} d^2(\mathbf{x}_i,\mathbf{x}_j) \leq 1 \\ & A \succeq 0 \end{aligned} \tag{17}$$

Optimization of this problem can be solved by iteratively using gradient ascent on $g(A)$, followed by applying projections to $f(A)$.

3.3 Numerical example

In this section, we give a simple numerical example to evaluate our proposed semantic distances between vague concepts based on our definitions.

Humans have the ability to choose an appropriate word to describe the object's color, based on their subjective belief. In this sense, color is a vague concept and the border of two colors is difficult to defined by a specific numerical value. However, the method introduced in this paper can be used to quantize the differences between colors and make the agents have the ability to distinguish colors.

HSV (hue, saturation and value) is the most common representations of color model. Hue, saturation and value together describe a color. For simplicity, we only use labels including *"red(r)"*, *"orange(o)"* and *"yellow(y)"* to cover the range [0 90] of the H axis of HSV color space in this example. We conduct a simple example to ask ten students to label ten data points uniformly drawn from the H axis covering the range [0 90]. This method results in a side information that to be used to learn the appropriateness measure function for each label using SVR (see Sect. 3.2). The learned appropriateness measure functions are shown in Fig. 4. The parameter p in Definition 6 is fixed to 2.

In this example, we pick five data instances: $H = 15$, $H = 40$, $H = 65$, $H = 75$, $H = 80$. Six label expressions are also picked including three basic labels: "red," "orange,"

Table 1 Distances between data instance on H values

	$H = 15$	$H = 40$	$H = 65$	$H = 75$	$H = 80$
$H = 15$	0	0.8952	0.9232	1.0561	1.1624
$H = 40$	0.8952	0	0.5520	0.9691	1.1358
$H = 65$	0.9232	0.5520	0	0.4078	0.5788
$H = 75$	1.0561	0.9691	0.4078	0	0.1676
$H = 80$	1.1624	1.1358	0.5788	0.1676	0

"yellow" and three label expressions: "red$\vee$orange," orange$\wedge$ yellow, $\neg$red, generated by connecting basic labels using standard logical connection. We calculate the semantic distances between them. Table 1 shows the distances between instances by calculating using Definition 6. Table 2 shows the distances between instance and label expression by calculating using Definition 7. The distances between label expressions are also shown in Table 3. From these tables, we can see that our defined distances between label expressions are symmetric and have the ability to reflect the semantic divergences between vague color concepts. For example, red $\vee$ orange means both red and orange are appropriate to describe some instance, while orange means it is appropriate to describe the instance using orange. In other words, some instances which can be appropriately described as orange also can be labeled red $\vee$ orange, but not vice versa. Thus, $d(\text{orange}, \text{red} \vee \text{orange}) = 0.3261$. From the result table, we also got that $d(\text{orange}, \text{orange} \wedge \text{yellow}) > d(\text{orange}, \text{red} \vee \text{orange})$ because orange and red$\vee$orange have larger semantic divergence. In addition, the distances between multidimensional label expressions are not shown here and they will be illustrated in Sect. 5.

4 Clustering based on semantic distance with side information

Last example (see 3.3) shows that the distances between objects including numerical data and vague concepts represented by label expressions can be easily measured using our defined semantic distances. One application of our semantic distances is "clustering with side information." In this section, we design clustering algorithms by combining our proposed semantic distances and classical clustering algorithms together. "Side information" here is a prior knowledge about the data. Specifically, we pick several pares of objects from each class to form the side information $\mathbb{S} = \{S_1, S_2, \ldots, S_K\}$, and each $\mathbf{x}_i \in S_p$, $p = 1, 2, \ldots, K$ belongs to the cluster p. We will learn the appropriateness measures of all basic labels with this side information. Furthermore, the distance metric learning for multidimensional label expressions is also based on the side information.

K-medoids (Chen et al. 2009) and spectral clustering (Ng et al. 2009) are two widely used unsupervised learning algorithms to solve clustering problems. The objects clustered by these algorithms must be homogeneous. For example, the objects only include images in image clustering problem. All objects are indicated as vectors, and then, Euclidean distance (or other distance measures) is used to measure their distances which determine the objects' cluster labels. These classical algorithms cannot deal with mixed objects including numerical data and vague concepts. In this work, we combine these classical clustering algorithms and our proposed semantic distances to design new clustering algorithms, which can not only cluster homogeneous objects but also have the ability to deal with mixed objects. The main differences between these classical clustering algorithms and our modified algorithms include: (1) To learn the parameters

Table 2 Distances between data instance and label expressions on H values

	Red	Orange	Yellow	Red $\vee$ Orange	Orange $\wedge$ Yellow	$\neg$ Red
$H = 15$	0.3470	0.9581	1.2752	0.3470	0.9425	1.2752
$H = 40$	1.2415	0.0730	1.3133	0.0730	0.6684	1.3133
$H = 65$	1.2554	0.5509	0.7559	0.5509	0.1052	0.7559
$H = 75$	1.2758	0.9583	0.3466	0.9583	0.3041	0.3466
$H = 80$	1.3405	1.1249	0.1801	1.1249	0.4718	0.1801

Table 3 Distances between label expressions on H values

	Red	Orange	Yellow	Red $\vee$ orange	Orange $\wedge$ yellow	$\neg$ Red
Red	0	1.3040	1.4143	0.3261	1.3161	1.4147
Orange	1.3040	0	1.3041	0.3261	0.6554	1.3041
Yellow	1.4143	1.3041	0	1.3316	0.6510	0
Red $\vee$ orange	0.3261	0.3261	1.3316	0	0.7956	1.3316
Orange $\wedge$ yellow	1.3161	0.6554	0.6510	0.7956	0	0.6512
$\neg$ Red	1.4147	1.3041	0	1.3316	0.6512	0

of our semantic distances, we need a small part of the objects tell us that they are semantically similar or dissimilar, while both of classical k-medoids algorithm and spectral clustering need no supervised information. (2) The work flows of algorithms remain the same, except that the distance metrics, such as Euclidean distance, are replaced by our semantic distances. Actually our semantic distances based on semantic labels can be applied to any other clustering models which need distance metric to evaluate the dissimilarities between objects.

We first introduce the method to get training set for learning appropriateness measure functions and metric learning for multidimensional label expressions. Suppose we are given a set of instances $\mathbf{x}_i, i = 1, 2, \ldots, N$, and each one has P properties and x_i^j denotes the value of the jth property of $\mathbf{x}_i$. The side information is also given as $\mathbb{S} = \{S_1, S_2, \ldots, S_K\}$, where each instance $\mathbf{x}_i \in S_p$ belongs to the cluster p. We cover each dimensional property with K cluster labels $\{L_1, L_2, \ldots, L_K\}$. For label L_m on jth-dimensional property, we use instances in S_m to learn appropriateness measure function. Suppose Ω^j is the value range of jth-dimensional property, and it is uniformly divided into T (we call it property granularity) pieces $R_1^j, R_2^j, \ldots, R_T^j$. C_i denotes that how many instances in S_m fall into R_i^j. Thus, we get training sets to learn appropriateness measures for each label from the side information:

$$\{(x_i^j, \hat{\mu}_{L_m}(x_i^j)) : \mathbf{x}_i \in S_m, \quad j = 1, \ldots, P, \quad m = 1, \ldots, K\}$$

where

$$\hat{\mu}_{L_m}(x_i^j) = \frac{C_k}{\max\{C_n, n = 1, 2, \ldots, T\}}, \quad \text{where } x_i^j \in R_k^j$$

Voted by instances in side information, we get a histogram for each label and this histogram is just the training set for this label (see Eq. 10 in Sect. 3.2.1). This approach is demonstrated by Fig. 5. Then we can learn the appropriateness measure function of each label using the method introduced in Sect. 3.2.1.

Similarly, metric learning for $MLEs$ is also based on the side information. The similar instance set $\mathcal{S}$ and dissimilar instance set $\mathcal{D}$ can formed from side information $\mathbb{S}$ as follows.

$$\mathcal{S} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i, \mathbf{x}_j \in S_p, p = 1, 2, \ldots, K\}$$
$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \in S_p, \mathbf{x}_j \in S_q, p \neq q\}$$

Thus, we can learn the distance metric for MLEs according to the method introduced in Sect. 3.2.2.

Our proposed semantic distances can be easily employed by classical clustering algorithms and make them be able to deal with mixed objects. In this paper, we try two popular clustering algorithms including k-medoids and spectral clustering. The pseudo-code of our proposed clustering algorithms are shown in Tables 4 and 5. Being different from classical clustering algorithms, the input data of our proposed clustering methods $\Theta = \{\mathbf{O}_i : i = 1, \ldots, N\}$ contain not only numerical data instances but also vague concepts. Our methods can cluster numerical data instances (which also can be clustered by classical clustering algorithms) or vague concepts alone, also have the ability to cluster numerical data instances and vague concepts simultaneously. For convenience, we call our methods hybrid clustering when they are used to cluster mixed objects, and traditional clustering otherwise. Figure 6 gives a schematic diagram of our proposed clustering methods, where vague concepts presented by ellipses correspond to label expressions or other granularity, while data instances are presented by dots. Such mixed objects can be clustered by our clustering algorithm.

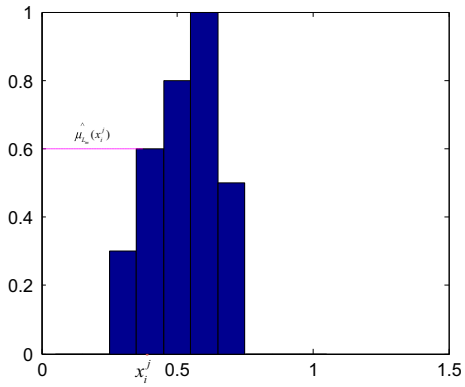


Fig. 5 Training set for learning appropriateness measure is a histogram voting by instances in side information

5 Experiments

In this section, we carry out several experiments to demonstrate the effectiveness of our proposed semantic distances and clustering algorithms. A dataset containing images and image labels is used to illustrate how our methods cluster mixed objects. We also evaluate our methods by clustering documents on two widely used data corpora and compare our results with some popular existing algorithms. All experiments are performed in Windows 7 environment on a computer with Intel core i5 CPU and 8 GB RAM and all the algorithms are coded in MATLAB.

5.1 Datasets

Corel images and labels We picked three classes of images from Corel image data set (Lavrenko et al. 2004; Carneiro et al. 2006; Zhang et al. 2012) and each class has one hundred images. First these images are resized into 192 × 128

Table 4 Semantic distance-based k-medoids clustering algorithm

Given: data set $\Theta = \{\mathbf{O}_i : i = 1, \ldots, N\}$ cluster number $K > 0$, and termination condition ε
Initialize counter $p = 0$ and each cluster center $\mathbf{c}_1$,..., $\mathbf{c}_K$ loop: $p++$
Step 1: For each object in Θ, determine the cluster $\mathbf{O}_i \leftarrow c_j$, if:
$d(\mathbf{O}_i, \mathbf{c}_j^{(p-1)}) = \min\{d(\mathbf{O}_i, \mathbf{c}_k^{(p-1)}) : k = 1, \ldots, K\}$
Step 2: Update the cluster centers $\mathbf{c}_i^{(p)}$:
$\mathbf{c}_i^{(p)} = \arg\min_{\mathbf{x} \in C_i} \{\sum_{\mathbf{O}_j \in \mathbb{C}_i} d(\mathbf{x}, \mathbf{O}_j)\}$
Until $\sum_{i=1}^{K} d(\mathbf{c_i}^{(p)}, \mathbf{c_i}^{(p-1)}) < \varepsilon$

Table 5 Semantic distance-based spectral clustering algorithm

Given:	Data set $\Theta = \{\mathbf{O}_i : i = 1, \ldots, N\}$ and cluster number $K > 0$
Step 1:	Construct similarity graph
	Connect all objects, and weight all edges by $s_{ij} = \exp\{-d(\mathbf{O}_i, \mathbf{O}_j)^2/\sigma^2\}$
Step 2:	Compute the normalized Laplacian L
Step 3:	Compute the first m eigenvectors $u_1, \ldots, u_m$ of L and let $U \in \mathbb{R}^{N \times m}$ be the matrix containing the vectors $u_1, \ldots, u_m$ as columns
Step 4:	Construct matrix $T \in \mathbb{R}^{N \times m}$ by normalizing the rows of U, $t_{ij} = u_{ij}/(\sum_{k=1}^{m} u_{ik}^2)^{1/2}$
Step 5:	Let $y_i \in \mathbb{R}^m$ be the vector corresponding to the ith row of T
Step 6:	Clustering $(y_i)_{i=1,\ldots,N}$ with k-means algorithm into K clusters $C_1, \ldots, C_K$

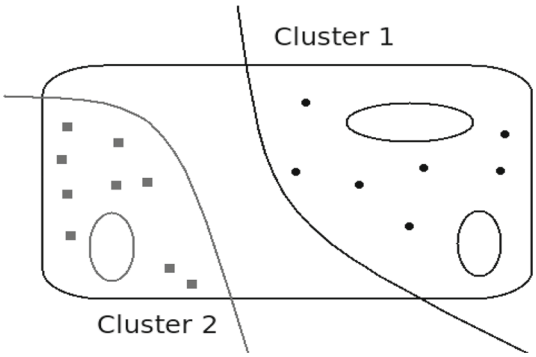


Fig. 6 A schematic illustration of hybrid clustering algorithm. Ellipses represent the vague concepts and dots are for data instances

pixels. Then three labels are artificially designed: "grass," "polar bear" and "sunset." These labels can be considered as vague concepts which describe the contents of images. Before clustering, several images are picked from each class to compose the side information.

Reuters-21578 This dataset[2] is a collection of 21,578 documents falling into over 135 thematic categories, collected from news published by Reuters newswire. It is widely used as a benchmark dataset for document clustering (Deng et al. 2005; Bharti and Singh 2016). In this paper, we randomly select 15 categories of documents. We discard those documents with multiple category labels. Finally we get 2408 documents in total.

Fig. 7 Example images of vague concept 'sunset'

TDT2 It is the abbreviation of "Nist Topic Detection and Tracking corpus",[3] which is collected in 1998 from six sources including newswires, radio programs and television programs, and consists of 11,201 documents in 96 categories. It is also widely used as a benchmark dataset for document clustering (Deng et al. 2005). We randomly select 3080 documents from 15 categories. The documents with multiple category labels are also removed.

5.2 Setup and Performance of hybrid clustering

By modeling vagueness with label semantics, data instances and vague concepts can be embedded into appropriateness space in which distance between data instances and vague concepts can be measured. Based on this property, we design hybrid clustering algorithms to cluster mixed object, such as images and labels, simultaneously. Here, we consider labels such as "sunset" as vague concept, because we cannot give a definition of sunset to cover all the images that show sunset landscape. For example, all the three images in Fig. 7 can be

[2] Reuters-21578 is available at http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[3] TDT2 is available at http://www.itl.nist.gov/iad/mig/publications/proceedings/darpa99/html/tdt110/tdt110.htm.

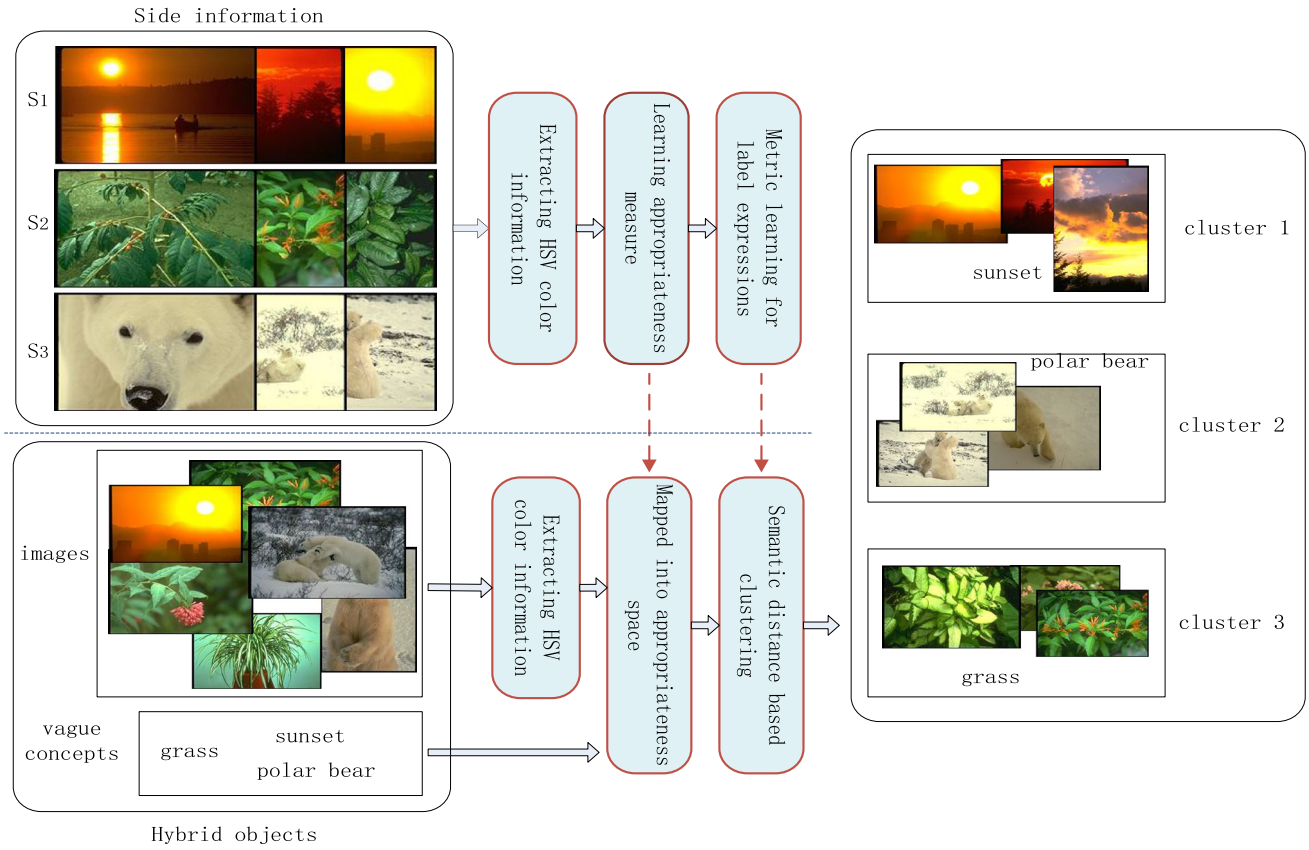


Fig. 8 Flowchart of the images and labels clustering by semantic distance-based k-means

viewed as sunset landscape. But sunset is not limited to these images. In other words, the label sunset cannot be defined using any one of these images. Taken in this sense, sunset is a vague concept. In our hybrid clustering, vague concepts are clustered as well as data instances.

In order to show the ability of our hybrid clustering algorithms, we apply them on Corel images and labels dataset to cluster images and labels simultaneously. Figure 8 shows the flowchart of clustering images and labels. For each image in side information, the average hue, saturation and value are extracted to be considered as three properties of each image. Each property is covered by the same three labels: "grass(g)," "polar bear(p)" and "sunset(s)." We then learn the appropriateness measures for each label on each property as shown in Fig. 9. The SVR we used in this experiment is based on LibSVM[4] developed by Chang and Lin (2011). Finally we learn distance metric for label expressions using the method introduced in Sect. 3.2.2 based on side information. We learned both the diagonal and full A.

In the clustering stage, the average HSV color of all pixels of an image is extracted and then mapped into appropriateness space. Thus, each image can be represented as a matrix as shown in Fig. 10. In this experiment, we study the effectiveness of amount of side information, property granularity and the parameter p in Definition 6 (we call it norm order). Since existing classical clustering algorithms cannot cluster mixed objects, we only consider our proposed algorithms including:

- Semantic distance-based k-medoids (SD k-medoids): k-medoids based on the learned semantic distance from $\mathcal{S}$ and $\mathcal{D}$ (including diagonal and full A).
- Semantic distance-based constrained k-medoids (SDC k-medoids): k-medoids based on the learned semantic distance from $\mathcal{S}$ and $\mathcal{D}$, but subject to pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ (including diagonal and full A).
- Semantic distance-based spectral clustering (SD spectral clustering): spectral clustering based on the learned semantic distance from $\mathcal{S}$ and $\mathcal{D}$ (including diagonal and full A).
- Semantic distance-based constrained spectral clustering (SDC spectral clustering): spectral clustering based on the learned semantic distance from $\mathcal{S}$ and $\mathcal{D}$, but subject to pairs $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S}$ (including diagonal and full A).

[4] LibSVM is available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

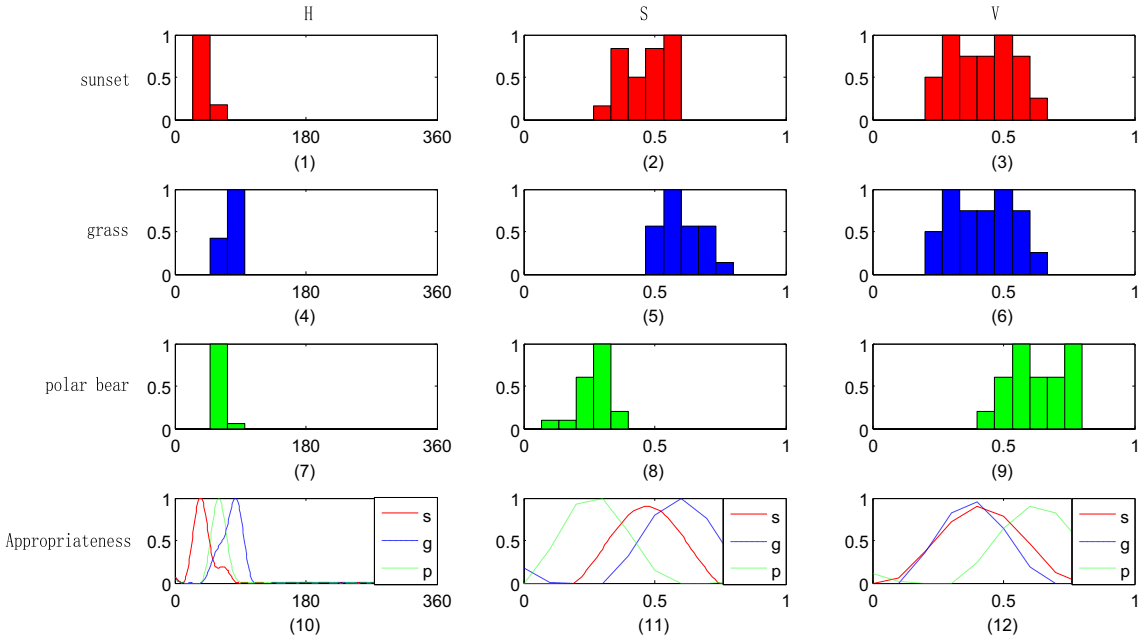
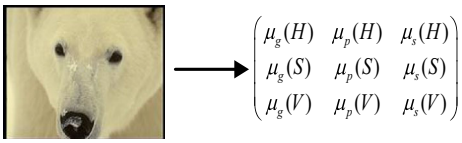


Fig. 9 Histograms (1)–(3) are the training sets for label "sunset" on the properties: Hue, Saturation and Value, respectively. Histogram (4)–(6) are the training sets for label "grass" on the properties: hue, saturation and value, respectively. Histogram (7)–(9) are the training sets for label "polarbear" on the properties: hue, saturation and value, respectively. Image (10) illustrates appropriateness measures of three labels on Hue property, learned from histograms (1) (4) (7). Image (11) illustrates appropriateness measures of three labels on Saturation property, learned from histograms (2) (5) (8). Image (12) illustrates appropriateness measures of three labels on Value property, learned from histograms (3) (6) (9)



$$\begin{pmatrix} \mu_g(H) & \mu_p(H) & \mu_s(H) \\ \mu_g(S) & \mu_p(S) & \mu_s(S) \\ \mu_g(V) & \mu_p(V) & \mu_s(V) \end{pmatrix}$$

Fig. 10 Image can be represented as a matrix after mapped into appropriateness space. $\mu_g(H)$ is the appropriateness of describing image's Hue using label "grass". The rest elements in matrix can be explained by such analogy

In this experiment, we evaluate and compare the performances of these algorithms by clustering accuracy (AC), which compares the predicted labels of objects with their ground-truth labels. Suppose l_i and h_i to be the predicted cluster label and ground-truth label of object x_i, respectively, the clustering accuracy is defined as follows:

$$\mathrm{AC} = \frac{\sum_{i=1}^{n} I(h_i == \mathrm{map}(l_i))}{n} \tag{18}$$

where n is the total number of objects (images and labels) to be clustered, I(expression) is the indicator function whose value equals one if the *express* is true, equals zero otherwise. map(l_i) is a mapping function which maps predicted label l_i to the ground-truth label and we use the Kuhn–Munkres algorithm (Lovasz and Plummer 1986). All experiments are repeated 20 times, and the mean clustering accuracies are reported.

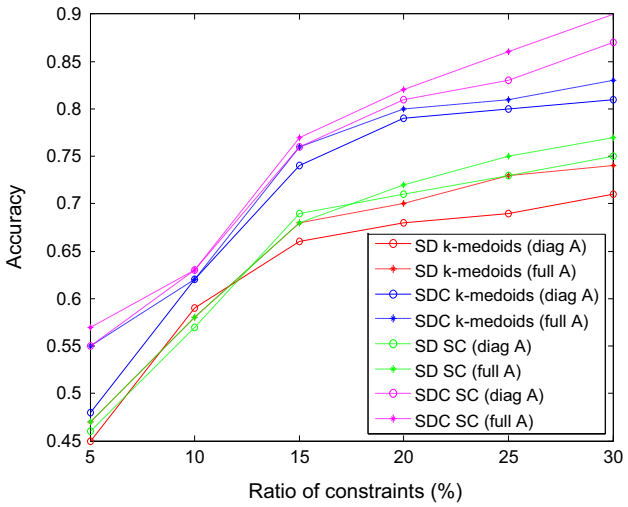


Fig. 11 Clustering accuracy versus side information. The fraction of all instances that are randomly sampled to be included in $\mathbb{S}$ is given by the x-axis, and we fix the parameters $T = 20$, $p = 2$

We first test the clustering performance using different amounts of side information, with fixed property granularity ($T = 20$) and fixed norm order ($p = 2$), as shown in Fig. 11.

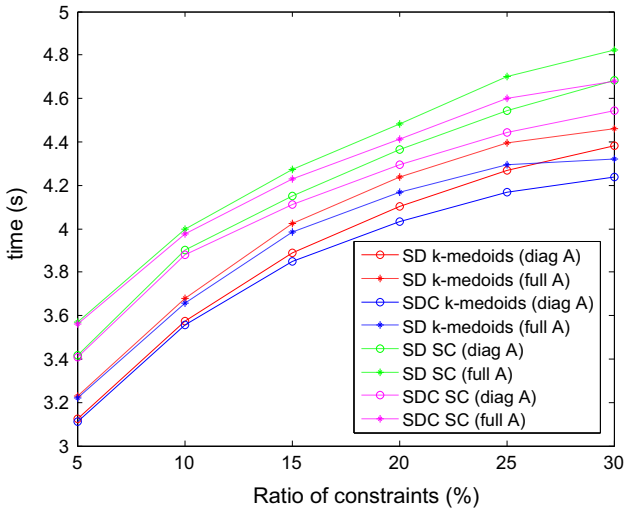


Fig. 12 Time consumption vs. side information. The fraction of all instances that are randomly sampled to be included in $\mathbb{S}$ is given by the x-axis, and we fix the parameters $T = 20$, $p = 2$

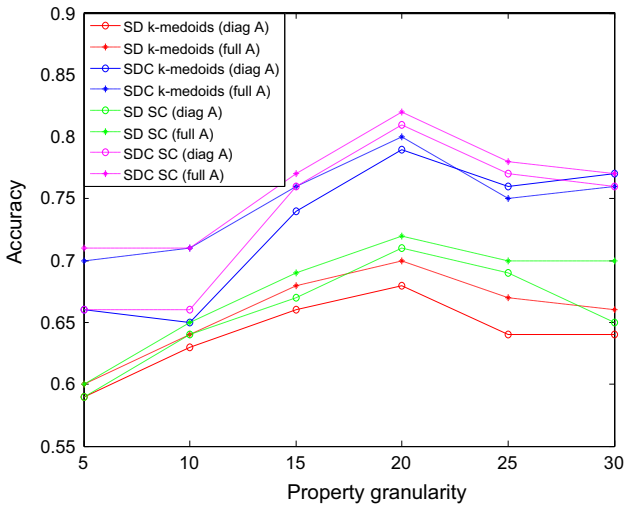


Fig. 13 Clustering accuracy versus property granularity. Here the ratio of constraint is fixed on 0.2, and norm order is 2

We also report the time consumption of learning with different amount of side information as shown in Fig. 12. By examining these figures, we can see clustering performances of all kinds of algorithms improved with increasing amount of side information and time consumption also rise. Then we fix the ratio of constraints on 0.2, fix the norm order on 2, choose the property granularity T from {5, 10, 15, 20, 25, 30} and compare the results shown in Fig. 13. It indicates that property granularity $T = 20$ outperforms all the other values of T for these images and labels. Finally we test the impact of norm order on clustering accuracy. We try the norm orders $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ one by one, while we fix the property granularity and ratio of constraint to be 20 and 0.2, respectively. Results are shown in Fig. 14. It shows that, for

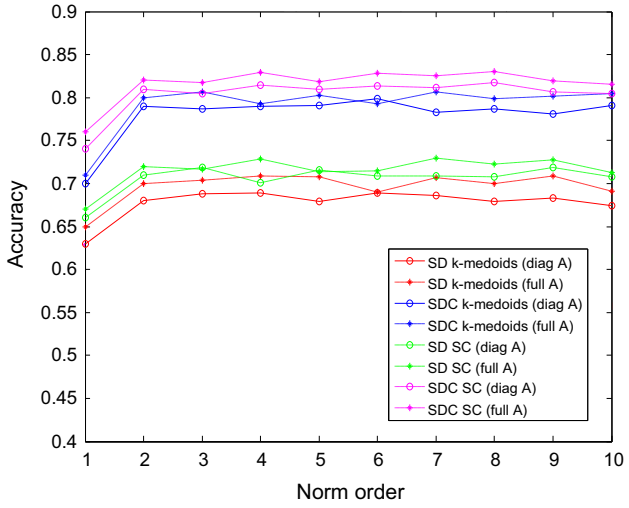


Fig. 14 Clustering accuracy versus norm order. Here the ratio of constraint is fixed on 0.2, and property granularity is 20

all algorithms, the clustering accuracies with norm order 1 are much lower than those when the norm order is more than one. The clustering performance become relatively stable when the norm order is more than one. Figure 15 shows several samples drawn from images and labels clustered by our proposed semantic distance-based k-medoids algorithm. Other algorithms we proposed have similar results. Most images and all vague concepts *"sunset"*, *"grass"* and *"polarbear"* are correctly classified, except that image in the green box is wrongly clustered into the *sunset* cluster, while its content is two polar bears. The reason is that HSV color of the image in green box is much more similar to the images in sunset cluster than that in polar bear cluster. The reason is that color feature is only one useful feature to describe images, but it is not enough to reflect the semantics of an image completely. Actually, high-level image features, such as SIFT and LBP, can convey more accurate information about the images' contents. These features can also be integrated in our approach.

5.3 Setup and performance of traditional clustering

In this section, we use our proposed clustering methods to solve document clustering problem and compare with some recently proposed algorithms. For fairly comparison, we test all the algorithms on the Reuters-21578 and TDT2 datasets. To cluster documents, we should preprocess the documents. First we remove the stop words (such as a, an, is and the) from each document. In this paper, stop words are from the Lextek stop word list,[5] which contains 429 words. Second

[5] http://www.lextek.com/manuals/onix/stopwords1.html.

Fig. 15 Result samples of hybrid clustering of images and labels. Most images and all labels are correctly classified, except that the image in green box is wrongly clustered

we use TF-IDF scheme (Wu et al. 2008) to map textual document into vector representation. TF-IDF is mathematically formulated as follows:

$$\text{tfidf}_{ij} = \begin{cases} \text{tf}_{ij} \times \ln(\frac{n}{df_j}), & \text{if} \quad \text{tf}_{ij} \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where tfidf_{ij} is the TF-IDF value of the jth term in the ith document, tf_{ij} is the frequency of the jth term in the ith document, n is the total number of documents in the dataset and df_j is the number of documents in the dataset who have the jth term. In the third preprocessing step, we use PCA (Jolliffe 2005) to reduce the dimensionality of vector representation of each document, in order to reduce computational requirement. In this experiment, we compare our proposed algorithms SD k-medoids with full A (SD k-medoids), SDC k-medoids with full A (SDC k-medoids), SD spectral clustering with full A (SD SC) and SDC spectral clustering with full A (SDC SC) with basic k-medoids (Chen et al. 2009), basic spectral clustering(SC) (Ng et al. 2009), basic k-means (MacQueen 1967), PSO+k-means (Xiaohui et al. 2005), LPI (Deng et al. 2005), basic ABC (Karaboga 2005), GABC (Zhu and Kwong 2010) and CGABC (Bharti and Singh 2016). The parameters, including ratio of constraints, property granularity and norm order of our proposed algorithms, are set to be 20%, 20, 2, respectively. The parameter setting for ABC, GABC and CGABC follows Bharti and Singh (2016). Cosine distance is used as the distance metric in basic k-medoids, k-means and spectral clustering. In this experiment, we adopt the evaluation metrics mean precision (P), mean recall (R), and F score (F), which are widely used in text categorization (Figueiredo et al. 2011). For category i, precision P_i measures the total number of documents correctly clustered into category i (TP_i) to the total number of documents clustered into category i ($\text{TP}_i + \text{FP}_i$). Recall R_i is the total number of documents correctly clustered into category i (TP_i), divided by the total number of documents in category i ($\text{TP}_i + \text{FN}_i$). Then we calculate the mean precision and recall. F score is the trade-off between mean precision and recall which combine them together. P, R, F are calculated using equation 20, 21, 22, respectively (C is the number of categories). Furthermore, we also compare their computational time. The experiment is repeated 20 times, and the average, best and worst performances are reported.

$$P = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \tag{20}$$

$$R = \frac{1}{C}\sum_{i=1}^{C}\frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \tag{21}$$

$$F = \frac{2 * P * R}{P + R} \tag{22}$$

Tables 6 and 7, respectively, give the comparative results of our methods and some existing algorithms on Reuters-21578 and TDT2. It is obvious that all of our methods significantly outperform most of the existing algorithms including k-medoids, spectral clustering, k-means, PSO + k-means, LPI, ABC, GABC on the two datasets. The best results are obtained by our method semantic distance-based spectral clustering (SDC SC), which outperform the state-of-the-art algorithm CGABC. Being different from the existing algorithms which need no prior knowledge of the data to be clustered, our methods need the side information (a small part of the dataset) to learn the appropriateness measure functions and distance for multidimensional label expressions. Based on this side information, our methods remarkably improve the clustering performance and reduce the computational time.

6 Conclusion

In this paper, we model vague concept with label semantics which is a framework of computing with words and design a series of novel semantic distances to evaluate similarities between data instances and vague concepts. Being different from the existing fuzzy set distances, the distances defined

Table 6 Statistical analysis (A: average, B: best, W: worst) of methods in terms of P, R, F and computational time over 20 independent runs on Reuters-21578

Method	P			R			F			Computational time (ms)		
	A	B	W	A	B	W	A	B	W	A	B	W
k-medoids	0.352	0.398	0.311	0.237	0.262	0.194	0.283	0.313	0.235	416,122	410,354	421,090
SC	0.405	0.442	0.378	0.248	0.270	0.205	0.307	0.331	0.266	494,262	486,218	501,886
k-means	0.351	0.386	0.321	0.238	0.266	0.201	0.283	0.311	0.247	398,065	390,866	402,113
PSO + k-means	0.398	0.412	0.376	0.248	0.279	0.227	0.305	0.336	0.284	425,675	421,008	431,672
LPI	0.412	0.432	0.399	0.268	0.290	0.238	0.324	0.347	0.293	496,221	490,655	501,101
ABC	0.422	0.435	0.410	0.271	0.299	0.257	0.330	0.351	0.314	7,866,340	7,812,098	7,898,531
GABC	0.440	0.478	0.411	0.295	0.308	0.279	0.353	0.374	0.332	7,908,680	7,899,405	7,999,809
CGABC	0.558	0.579	0.536	0.326	0.342	0.308	0.411	0.432	0.391	8,018,072	7,999,867	8,027,909
SD k-medoids	0.489	0.501	0.473	0.298	0.315	0.284	0.370	0.387	0.356	2,567,098	2,558,099	2,570,901
SDC k-medoids	0.539	0.551	0.490	0.324	0.329	0.299	0.404	0.415	0.370	2,568,909	2,553,658	2,580,980
SD SC	0.541	0.558	0.538	0.318	0.322	0.299	0.400	0.412	0.383	3,048,644	3,040,989	3,050,912
SDC SC	0.588	0.603	0.569	0.341	0.356	0.321	0.431	0.449	0.409	3,047,829	3,040,878	3,050,001

Table 7 Statistical analysis (A: average, B: best, W: worst) of methods in terms of P, R, F and computational time over 20 independent runs on TDT2

Method	P			R			F			Computational time (ms)		
	A	B	W	A	B	W	A	B	W	A	B	W
k-medoids	0.405	0.442	0.379	0.298	0.312	0.269	0.343	0.364	0.312	501,283	487,103	510,982
SC	0.431	0.458	0.402	0.290	0.316	0.279	0.346	0.374	0.331	589,980	580,419	599,012
k-means	0.401	0.436	0.388	0.290	0.314	0.265	0.336	0.370	0.313	497,091	480,154	508,701
PSO + k-means	0.412	0.436	0.398	0.291	0.314	0.270	0.341	0.366	0.326	520,908	514,120	531,230
LPI	0.452	0.474	0.418	0.299	0.308	0.276	0.359	0.371	0.328	539,080	530,190	547,045
ABC	0.443	0.476	0.409	0.289	0.296	0.261	0.349	0.360	0.316	9,451,880	9,408,112	9,510,976
GABC	0.496	0.512	0.470	0.312	0.335	0.298	0.383	0.406	0.362	9,678,210	9,590,912	9,739,080
CGABC	0.601	0.624	0.587	0.398	0.416	0.373	0.478	0.499	0.452	9,806,787	9,718,769	9,918,702
SD k-medoids	0.585	0.606	0.561	0.324	0.355	0.307	0.417	0.445	0.393	2,890,782	2,887,609	2,986,520
SDC k-medoids	0.603	0.630	0.589	0.331	0.354	0.319	0.427	0.449	0.414	2,889,076	2,830,951	2,974,713
SD SC	0.598	0.610	0.573	0.325	0.354	0.319	0.421	0.447	0.414	2,040,960	2,030,517	2,045,097
SDC SC	0.623	0.644	0.608	0.412	0.430	0.399	0.496	0.513	0.480	2,047,034	2,038,471	2,050,573

in linguistic label space focus on the semantic divergence between labels.

Example studies on a numerical example showed that the semantic distances defined in this paper could evaluate the semantic divergences between vague concepts. Furthermore, we proposed two novel semantic distance-based clustering algorithms with side information. These clustering algorithms can be used not only to cluster data instances, but also to cluster hybrid objects including data instances and vague concepts. The image and label clustering experiment demonstrated that our approach is able to cluster mixed objects including numerical data and linguistic labels. Experiments of document clustering on two benchmark datasets show that our methods improve the clustering performance and reduce the computational time, compared with several existing algorithms.

As one of our future work, we will find more application areas for these semantic distances and clustering algorithms based on sematic distances. We will also go to investigate new rule-learning methods by combining the hybrid clustering algorithm to help agent generate appropriate label expressions to describe objects. Another future work is to investigate how to incorporate sentic computing (Cambria and Hussain 2012; Cambria 2012), which supplies a number of techniques to mining people's opinions, with label semantics. Combining sentic computing and label semantics may help artificial intelligence systems learn knowledge from the huge amount of unstructured information distributed on Web.

Acknowledgements This work is supported by the Natural Science Foundation of China (Grant Nos. 61572162 and 61272188) and the Zhejiang Provincial Key Science and Technology Project Foundation (No. 2017C01010).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Bharti K, Singh P (2016) Chaotic gradient artificial bee colony for text clustering. Soft Comput 20(3):1113–1126

Bishop M (2006) Pattern recognition and machine learning. Springer, Berlin

Cambria E (2012) Sentic computing for socal media marketing. Multimed Tools Appl 59(2):557–577

Cambria E, Hussain A (2012) Sentic computing: techniues, tools, and applications. Springer, Berlin

Carneiro G, Chan A, Moreno P, Vasconcelos N (2006) Supervised learning of semantic classes for image annotation and retrieval. IEEE Trans PAMI 29(3):394–410

Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27

Chen Y, Garcia E, Gupta M, Rahimi A, Cazzanti L (2009) Similarity-based classification: concepts and algorithms. J Mach Learn Res 10(2):747–776

Cortes C, Vapnik V (1995) Support vector networks. Mach Learn 20(3):273–297

Crosscombe M, Lawry J (2016) A model of multi-agent consensus for vague and uncertain beliefs. Adapt Behav 24(4):249–260

Daniel R, Lawry J, Rico-Ramirez A, Clukie D (2007) Classification of weather radar images using linguistic decision trees with conditional labelling. In: FUZZ-IEEE, pp 1–6

David A (2005) Statistical models: theory and practice. Cambridge University Press, Cambridge

Deng C, He X, Han J (2005) Document clustering using locality preserving indexing. IEEE Trans Knowl Data Eng 17(12):1624–1637

Figueiredo F, Rocha L, Couto T, Salles T, Goncalves M (2011) Word co-occurrence features for text classification. Inf Syst 36(5):843–858

Francisco A, Martinez J, Aguilar C, Roldon C (2016) Estimation of a fuzzy regression model using fuzzy distances. IEEE Trans Fuzzy Syst 24(2):344–359

Goldberger J, Hinton G, Roweis S, Salakhutdinov R (2005) Neighbourhood components analysis. In: NIPS, pp 513–520

Gu B, Sheng VS (2016) A robust regularization path algorithm for v-support vector classification. IEEE Trans Neural Netw Learn Syst 1:1–8

Gu B, Sheng VS, Tay KY, Romano W, Li S (2015a) Incremental support vector learning for ordinal regression. IEEE Trans Neural Netw Learn Syst 26(7):1403–1416

Gu B, Sheng VS, Wang Z, Ho D, Osman S, Li S (2015b) Incremental learning for v-support vector regression. Neural Netw 67:140–150

Gu B, Sun X, Sheng VS (2016) Structural minimax probability machine. IEEE Trans Neural Netw Learn Syst 28(7):1646–1656

H Druker CB (1997) Support vector regression machine. In: NIPS, pp 155–161

Guo H, Wang X, Wang L (2016) Delphi method for estimating membership function of uncertain set. J Uncertain Anal Appl 4(1):1–17

He H, Lawry J (2014) The linguistic attribute hierarchy and its optimisation for classification. Soft Comput 18(10):1967–1984

Janis V, Montes S (2007) Distance between fuzzy sets as a fuzzy quantity. Acta Univ Matthiae Belii Ser Math 14:41–49

Jolliffe I (2005) Principal component analysis. Wiley Online Library, Hoboken

Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. In: Technical report, Engineering faculty, Computer Engineering Department. Erciyes University Press, Erciyes

Lavrenko V, Manmatha R, Jeon J (2004) A model for learning the semantics of pictures. In: NIPS

Lawry J (2006) Modelling and reasoning with vague concepts. Springer, Berlin

Lawry J (2014) Probability, fuzziness and borderline cases. Int J Approx Reason 55(5):1164–1184

Lawry J, Tang Y (2009) Uncertainty modelling for vague concepts: a prototype theory approach. Artif Intell 173:1539–1558

Lewis M, Lawry J (2016) Hierarchical conceptual spaces for concept combination. Aritif Intell 237:204–227

Li D (2004) Some measures fo dissimilarity in intuitionistic fuzzy structures. J Comput Syst Sci 8:115–122

Hyung LK, Song KLYS (1994) Similarity measrue between fuzzy sets and between elements. Fuzzy Sets Syst 62:291–293

Lovasz L, Plummer M (1986) Matching theory. Budapest

MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley symposium on mathematical statistics and probability, pp 281–297

McCulloch J, Wagner C, Akckelin U (2013) Measuring the directional distance between fuzzy sets. In: UKCI 2013, the 13th annual workshop on computational intelligence, Surrey University, pp 38–45

Ng A, Jordan M, Weiss Y (2009) On spectral clustering: analysis and an algorithm. J Mach Learn Res 10(2):747–776

Nieradka G, Butkiewicz B (2007) A method for automatic membership function estimation based on fuzzy measures. Foundations of fuzzy logic and soft computing. Springer, Berlin, Heidelberg, pp 451–460

P Groenen UK, Rosmalen JV (2007) Fuzzy clustering with minkowski distance function. In: Advances in fuzzy clustering and its applications, pp 53–68

Pappis C, Karacapilidis N (1993) A comparative assessment of measures of similarity of fuzzy values. Fuzzy Sets Syst 56:171–174

Qin Z, Lawry J (2005) Decision tree learning with fuzzy labels. Inf Sci 172(1–2):91–129

Qin Z, Lawry J (2008) LFOIL: Linguistic rule induction in the label semantic framework. Fuzzy Sets Syst 159(4):435–448

Qin Z, Tang Y (2014) Uncertainty modeling for data mining: a label semantics approach. Springer, Berlin

Rosch E (1973) Natural categories. Cogn Psychol 4:328–350

Rosch E (1975) Cognitive representation of semantic categories. J Exp Psychol 104:192–233

Rosmalen JV (2006) Fuzzy clustering with minkowski distance. In: Econometric, pp 53–68

Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323–2326

Medasani S, Kim J, Krishnapuram R (1998) An overview of membership function generation techniques for pattern recognition. Int J Approx Reason 19:391–417

Scott J (2012) Illusions in regression analysis. Int J Forecast 28(3):689

Smola A, Scholkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222

Szmidt E, Kacprzyk J (2000) Distances between intuitionistic fuzzy sets. Fuzzy Sets Syst 114:505–518

Turnbull O, Lawry J, Lowengerg M, Richards A (2016) A cloned linguistic decision tree controller for real-time path planning in hostile environments. Fuzzy Sets Syst 293:1–29

V Srivastava, Tripathi BK, Pathak VK (2011) An evolutionaru fuzzy clustering with minkowski distances. In: International conference on neural information processing, pp 753–760

Vapnik V (1998) Statistical learning theory. Wiley, Hoboken

Victor S, Semyon V (2006) A theoretical introduction to numerical analysis. CRC Press, Boca Raton

Weinberger K, Saul L (2009) Distance metric learning for large margin nearest neighbor classification. J Mach Learn Res 10:207–244

Wu H, Luk R, Wong K, Kwok K (2008) Interpreting tf-idf term weights as making relevance decisions. ACM Trans Inf Syst 26(3):55–59

Xiaohui C, Potok T (2005) Document clustering analysis based on hybrid PSO+ k-means algorithm. J Comput Sci Special issue (April 15):27–33

Xing EP, Jordan MI, Russell SJ, Ng AY (2002) Distance metric learning with application to clustering with side-information. In: NIPS, pp 521–528

Zadeh L (1965) Fuzzy sets. Inf Control 8(3):335–353

Zadeh L (1975) The concept of linguistic variable and its application to approximate reasoning part 2. Inf Sci 4:301–357

Zadeh L (1996) Fuzzy logic = computing with words. IEEE Trans Fuzzy Syst 4:103–111

Zhang W, Qin Z, Tao W (2012) Semi-automatic image annotation using sparse coding. In: ICMLC

Zhang Y, Schneider J (2012) Maximum margin output coding. In: ICML

Zheng Y, Jeon B, Xu D, Wu QJ, Zhang H (2015) Image segmentation by generalized hierarchical fuzzy c-means algorithm. Neural Netw 28(2):961–973

Zhu G, Kwong S (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. Appl Math Comput 217(7):3166–3173