Classification with coupled distance based clustering

Mu Li
University of Technology, Sydeny
Australia
muli60@gmail.com

Abstract

The similarity between nominal objects is not straightforward, especially in unsupervised learning. This paper proposes coupled similarity metrics for nominal objects, which consider not only intra-coupled similarity within an attribute (i.e., value frequency distribution) but also inter-coupled similarity between attributes (i.e. feature dependency aggregation). Four metrics are designed to calculate the inter-coupled similarity between two categorical values by considering their relationships with other attributes. The theoretical analysis reveals their equivalent accuracy and superior efficiency based on intersection against others, in particular for large-scale data. Substantial experiments on extensive UCI data sets verify the theoretical conclusions. In addition, experiments of clustering based on the derived dissimilarity metrics show a significant performance improvement.

1 Introduction

Similarity analysis has been a problem of great practical importance in several domains, including data mining, for decades [?]. By defining certain similarity measures between attribute values, it gauges the strength of the relationship between two data objects: the more two objects resemble each other, the larger the similarity is [?].

When objects are described by numerical features, their similarity measures geometric analogies which reflect the relationship of data values. For instance, the values $10m$ and $12m$ are more similar than $10m$ and $2m$. A variety of similarity metrics have been developed for numerical data, such as Euclidean and Minkowski distances [?]. By contrast, the similarity analysis between records described by nominal variables has received much less attention. Heterogeneous Distances [?] and Modified Value Distance Matrix (*MVDM*) [?], for example, depict the similarity between categorical values in supervised learning. For unlabeled data, only a few works [?], including Simple Matching Similarity (*SMS*, which only uses 0s and 1s to distinguish similarities between distinct and identical categorical values) and Occurrence Frequency [?], discuss the similarity between nominal values. We illustrate the problem with these works and the challenge of analyzing similarity for categorical data below.

Taking the Movie data (Table **??**) as an example, six movie objects are divided into two classes with three nominal features: director, actor and genre. The *SMS* measure between directors "*Scorsese*" and "*Coppola*" is 0, but "*Scorsese*" and "*Coppola*" are very similar directors[1]. Another observation by following *SMS* is that the similarity between "*Koster*" and "*Hitchcock*" is equal to that between "*Koster*" and "*Coppola*"; however, the similarity of the former pair should be greater since it belongs to the same class G_2.

Both instances show that it is much more complex to analyze similarity between nominal variables than continuous data, and *SMS* and its variants fail to capture the genuine relationship between nominal values. With the increase of categorical data such as that derived from social networks, it is important to develop effective and efficient measures for capturing similarity between nominal variables.

Thus, we discuss the similarity for categorical values by considering data characteristics. Two attribute values are similar if they present analogous frequency distributions for one attribute [?]; this reflects the intra-coupled similarity within a feature. For example, two directors are very similar if they appear with almost the same frequency, such as "*Scorsese*" with "*Coppola*" and "*Koster*" with "*Hitchcock*". However, the reality is that the former director pair is more similar than the latter. To improve the accuracy of intra-coupled similarity, it is believed that the object co-occurrence probabilities of attribute values induced on other features are comparable [?]. To this end, the similarity between directors should also cater for the dependencies on other features such as "actor" and "genre" over all the movie objects, namely, the inter-coupled similarity between attributes. The coupling relationships between values and between attributes contribute to a more comprehensive understanding of object similarity [?]. No work that systematically considers both intra-coupled and inter-coupled similarities has been reported in the literature. This fact leads to the incomplete description of categorical value similarities, and apart from this, the similarity analysis on dependency aggregation is usually very costly.

In this paper, we propose a Coupled Object Similarity (*COS*) measure by considering both Intra-coupled and

[1] A conclusion drawn from a well-informed cinematic source.

Inter-coupled Attribute Value Similarities (*IaAVS* and *IeAVS*), which capture the attribute value frequency distribution and feature dependency aggregation with a high learning accuracy and relatively low complexity, respectively. We compare accuracies and efficiencies among the four proposed metrics for *IeAVS*, and come up with an optimal one from both theoretical and experimental aspects; we then evaluate our proposed measure with an existing metric on a variety of benchmark categorical data sets in terms of clustering qualities; and we develop a method to define dissimilarity metrics flexibly with our fundamental similarity building blocks according to specific requirements..

The paper is organized as follows. In Section 2, we briefly review the related work. Preliminary definitions are specified in Section 3. Section 4 proposes the coupled similarities, and the theoretical analysis is given in Section 5. We demonstrate the efficiency and effectiveness of *COS* in Section 6 with experiments. Finally, we end this paper in Section 7.

2 Problem Statement

A large number of data objects with the same features can be organized by an information table $S =< U, A, V, f >$, where $U = \{u_1, \cdots, u_m\}$ is composed of a nonempty finite set of data objects; $A = \{a_1, \cdots, a_n\}$ is a finite set of features; $V = \bigcup_{j=1}^{n} V_j$ is a set of all attribute values, in which V_j is the set of attribute values of feature $a_j (1 \le j \le m)$; and $f = \wedge_{j=1}^{n} f_j$ $(f_j : U \to V_j)$ is an information function which assigns a particular value of each feature to every object. For instance, Table **??** consists of six objects and three features, with $f_2(u_1) = B_1$ and $V_2 = \{B_1, B_2, B_3\}$.

Generally speaking, the similarity between two objects $u_{i_1}, u_{i_2} \in U$ is built on top of the similarities within their values $x, y \in V_j$ for all the features a_j. The basic concepts below are defined to facilitate the formulation for attribute value similarities, where $|H|$ is the number of elements in H.

Definition 2.1 *Given an information table* S*, three* ***Set Information Functions (SIFs)*** *are defined as* $f_j^* : 2^U \to 2^{V_j}$, $g_j : V_j \to 2^U$*, and* $g_j^* : 2^{V_j} \to 2^U$*. Specifically:*

$$f_j^*(\{u_{k_1}, \cdots, u_{k_t}\}) = \{f_j(u_{k_1}), \cdots, f_j(u_{k_t})\}, \quad (2.1)$$

$$g_j(x) = \{u_i | f_j(u_i) = x, 1 \le j \le n, 1 \le i \le m\}, \quad (2.2)$$

$$g_j^*(W) = \{u_i | f_j(u_i) \in W, 1 \le j \le n, 1 \le i \le m\}, \quad (2.3)$$

where $u_i, u_{k_1}, \cdots, u_{k_t} \in U$*, and* $W \subseteq V_j$*.*

These *SIF*s describe the relationships between objects and attribute values from different levels. For example, $f_2^*(\{u_1, u_2, u_3\}) = \{B_1, B_2\}$, $g_2(B_1) = \{u_1, u_2\}$ for value B_1, while $g_2^*(\{B_1, B_2\}) = \{u_1, u_2, u_3, u_6\}$ if given $W = \{B_1, B_2\}$.

Definition 2.2 *Given an information table* S*, its* ***Inter-information Function (IIF)*** $\varphi_{j\to k} : V_j \to 2^{V_k}$ *is defined:*

$$\varphi_{j\to k}(x) = f_k^*(g_j(x)). \quad (2.4)$$

This *IIF* $\varphi_{j\to k}$ is the composition of f_k^* and g_j. It obtains the kth attribute value subset for the corresponding objects, which are derived from the jth attribute value x. For example, $\varphi_{2\to 1}(B_1) = \{A_1, A_2\}$.

Definition 2.3 *Given an information table* S*, the* k*th attribute value subset* $W \subseteq V_k$*, and the* j*th attribute value* $x \in V_j$*, the* ***Information Conditional Probability (ICP)*** *of* W *with respect to* x *is* $P_{k|j}(W|x)$*:*

$$P_{k|j}(W|x) = \frac{|g_k^*(W) \bigcap g_j(x)|}{|g_j(x)|}. \quad (2.5)$$

Intuitively, when given all the objects with the jth attribute value x, *ICP* is the percentage of the common objects whose kth attribute values fall in subset W and jth attribute value is exactly x as well. For example, $P_{1|2}(\{A_1\}|B_1) = 0.5$.

All these concepts and functions are composed to formalize the so-called coupled interactions between categorical attribute values, as presented below.

3 Coupled Similarities

In this section, ***Coupled Attribute Value Similarity (CAVS)*** is proposed in terms of both intra-coupled and inter-coupled value similarities. When we consider the similarity between attribute values, "intra-coupled" indicates the involvement of attribute value occurrence frequencies within one feature, while the "inter-coupled" means the interaction of other features with this attribute. For example, the coupled value similarity between B_1 and B_2 concerns both the intra-coupled relationship specified by the repeated times of values B_1 and B_2: 2 and 2, and the inter-coupled interaction triggered by the other two features (a_1 and a_3).

Suppose we have the ***Intra-coupled Attribute Value Similarity (IaAVS)*** measure $\delta_j^{Ia}(x, y)$ and ***Inter-coupled Attribute Value Similarity (IeAVS)*** measure $\delta_j^{Ie}(x, y)$ for feature a_j and $x, y \in V_j$, then *CAVS* $\delta_j^A(x, y)$ is naturally derived by simultaneously considering both of them.

Definition 3.1 *Given an information table* S*, the* ***Coupled Attribute Value Similarity (CAVS)*** *between attribute values* x *and* y *of feature* a_j *is:*

$$\delta_j^A(x, y) = \delta_j^{Ia}(x, y) \cdot \delta_j^{Ie}(x, y) \quad (3.1)$$

where δ_j^{Ia} *and* δ_j^{Ie} *are IaAVS and IeAVS, respectively.*

3.1 Intra-coupled Interaction

According to [**?**], it is a fact that the discrepancy of attribute value occurrence times reflects the value similarity in terms of frequency distribution. Thus, when calculating attribute value similarity, we consider the relationship between attribute value frequencies on one feature, proposed as intra-coupled similarity in the following.

Definition 3.2 *Given an information table* S*, the* ***Intra-coupled Attribute Value Similarity (IaAVS)*** *between attribute values* x *and* y *of feature* a_j *is:*

$$\delta_j^{Ia}(x, y) = \frac{|g_j(x)| \cdot |g_j(y)|}{|g_j(x)| + |g_j(y)| + |g_j(x)| \cdot |g_j(y)|}. \quad (3.2)$$

In this way, different occurrence frequencies indicate distinct levels of attribute value significance. Gan et al. [**?**] reveal that greater similarity is assigned to the attribute value pair which owns approximately equal frequencies. The higher these frequencies are, the closer such two values are. Thus, function (3.2) is designed to satisfy these two principles. Besides, since $1 \leq |g_j(x)|, |g_j(y)| \leq m$, then $\delta_j^{Ia} \in [1/3, m/(m+2)]$. For example, in Table , both values B_1 and B_2 are observed twice, so $\delta_2^{Ia}(B_1, B_2) = 0.5$.

Hence, by taking into account the frequencies of categories, an effective measure (*IaAVS*) has been captured to characterize the value similarity in terms of occurrence times.

3.2 Inter-coupled Interaction

In terms of *IaAVS*, we have considered the intra-coupled similarity, i.e., the interaction of attribute values within one feature a_j. This does not, however, involve the couplings between other features $a_k (k \neq j)$ and feature a_j when calculating attribute value similarity. Accordingly, we discuss this dependency aggregation, i.e., inter-coupled interaction.

In 1993, Cost and Salzberg [**?**] proposed a powerful method, *MVDM*, for measuring the dissimilarity between categorical values. *MVDM* considers the overall similarities of classification of all objects on each possible value of each feature. The idea is that attribute values are identified as being similar if they occur with the same relative frequency for all classifications. In the absence of labels, the above measure is adapted to satisfy our target problem by replacing the class label with some other feature to enable unsupervised learning. We regard this interaction between features as inter-coupled similarity in terms of the co-occurrence comparisons of *ICP*. The most intuitive variant is *IRSP*:

Definition 3.3 *Given an information table S, the* ***Inter-coupled Relative Similarity based on Power Set (IRSP)*** *between attribute values x and y of feature a_j based on another feature a_k is:*

$$\delta_{j|k}^{P}(x, y) = \min_{W \subseteq V_k} \{2 - P_{k|j}(W|x) - P_{k|j}(\overline{W}|y)\}, \quad (3.3)$$

where $\overline{W} = V_k \backslash W$ is the complementary set of a set W under the complete set V_k.

In fact, two attribute values are closer to each other if they have more similar probabilities with other attribute value subsets in terms of co-occurrence object frequencies. In Table **??**, by employing (3.3), we want to get $\delta_{2|1}^{P}(B_1, B_2)$, i.e. the similarity between two attribute values B_1, B_2 of feature a_2 regarding feature a_1. Since the set of all attribute values of feature a_1 is $V_1 = \{A_1, A_2, A_3, A_4\}$, the number of all power sets within V_1 is 2^4, i.e., the number of the combinations consisting of $W \subseteq V_1$ and $\overline{W} \subseteq V_1$ is 2^4. The minimal value among them is 0.5, which indicates that similarity $\delta_{2|1}^{P}(B_1, B_2) = 0.5$.

This process shows the combinational explosion brought about by the power set needs to be considered when calculating attribute value similarity by *IRSP*. We therefore try to define three more similarities based on *IRSP* as follows.

Definition 3.4 *Given an information table S, the* ***Inter-coupled Relative Similarity based on Universal Set (IRSU), Join Set (IRSJ), and Intersection Set (IRSI)*** *between attribute values x and y of feature a_j based on another feature a_k are the following formulae respectively:*

$$\delta_{j|k}^{U}(x, y) = 2 - \sum_{w \in V_k} \max\{P_{k|j}(\{w\}|x), P_{k|j}(\{w\}|y)\}, \quad (3.4)$$

$$\delta_{j|k}^{J}(x, y) = 2 - \sum_{w \in \bigcup} \max\{P_{k|j}(\{w\}|x), P_{k|j}(\{w\}|y)\}, \quad (3.5)$$

$$\delta_{j|k}^{I}(x, y) = \sum_{w \in \bigcap} \min\{P_{k|j}(\{w\}|x), P_{k|j}(\{w\}|y)\}, \quad (3.6)$$

where $w \in \bigcup$ and $w \in \bigcap$ denote $w \in \varphi_{j \to k}(x) \bigcup \varphi_{j \to k}(y)$ and $w \in \varphi_{j \to k}(x) \bigcap \varphi_{j \to k}(y)$, respectively.

Each kth attribute value $w \in V_k$, rather than its value subset $W \subseteq V_k$, is considered to reduce computational complexity. In this way, ***IRSU*** is applied to compute similarity $\delta_{2|1}^{U}(B_1, B_2)$, and we get $\delta_{2|1}^{U}(B_1, B_2) = 0.5$. Since *IRSU* only concerns all the single attribute values rather than exploring the whole power set, it has solved the combinational explosion issue to a great extent. In ***IRSU***, ***ICP*** is merely calculated 8 times compared with 32 times by ***IRSP***, which leads to a substantial improvement in efficiency. Then with (3.5), the calculation of $\delta_{2|1}^{J}(B_1, B_2)$ is further simplified since $A_3 \notin \varphi_{2 \to 1}(B_1) \bigcup \varphi_{2 \to 1}(B_2)$. Thus, we obtain $\delta_{2|1}^{J}(B_1, B_2) = 0.5$, which reveals the fact that it is enough to compute ***ICP*** with $w \in V_1$ that belongs to $\varphi_{2 \to 1}(B_1) \bigcup \varphi_{2 \to 1}(B_2)$ instead of all the elements in V_1. From this perspective, ***IRSJ*** reduces the complexity further when compared with *IRSU*. Based on ***IRSU***, an alternative ***IRSI*** is considered. For example, with (3.6), the calculation of $\delta_{2|1}^{I}(B_1, B_2)$ is once again simplified since only $A_2 \in \varphi_{2 \to 1}(B_1) \bigcap \varphi_{2 \to 1}(B_2)$. Then, we easily get $\delta_{2|1}^{I}(B_1, B_2) = 0.5$. In this case, it is sufficient to compute ***ICP*** with $w \in V_1$ which only belongs to $\varphi_{2 \to 1}(B_1) \bigcap \varphi_{2 \to 1}(B_2)$. It is trivial that the cardinality of intersection $\bigcap$ is no larger than that of join set $\bigcup$. Thus, *IRSI* is further more efficient than *IRSU* due to the reduction of intra-coupled relative similarity complexity.

Intuitively speaking, it is a fact that *IRSI* is the most efficient of all the proposed inter-coupled relative similarity measures: *IRSP*, *IRSU*, *IRSJ*, *IRSI*. In addition, all four measures lead to the same similarity result, such as 0.5.

According to the above discussion, we can naturally define the similarity between the jth attribute value pair (x, y) on top of these four optional measures by aggregating all the relative similarities on features other than attribute a_j.

Definition 3.5 *Given an information table S, the* ***Inter-coupled Attribute Value Similarity (IeAVS)*** *between attribute values x and y of feature a_j is:*

$$\delta_j^{Ie}(x, y) = \sum_{k=1, k \neq j}^{n} \alpha_k \delta_{j|k}(x, y), \quad (3.7)$$

where α_k is the weight parameter for feature a_k, $\sum_{k=1}^{n} \alpha_k = 1$, $\alpha_k \in [0, 1]$, and $\delta_{j|k}(x, y)$ is one of the inter-coupled relative similarity candidates.

Accordingly, we have $\delta_j^{Ie} \in [0, 1]$, then $\delta_j^{A} = \delta_j^{Ia} \cdot \delta_j^{Ie} \in [0, m/(m+2)]$ since $\delta_j^{Ia} \in [1/3, m/(m+2)]$.

In Table **??**, for example, $\delta_2^{Ie}(B_1, B_2) = 0.5 \cdot \delta_{2|1}(B_1, B_2) + 0.5 \cdot \delta_{2|3}(B_1, B_2) = (0.5 + 0)/2 = 0.25$ if $\alpha_1 = \alpha_3 = 0.5$ is taken with equal weight. Furthermore, coupled attribute value similarity (3.1) is obtained as $\delta_2^A(B_1, B_2) = \delta_2^{Ia}(B_1, B_2) \cdot \delta_2^{Ie}(B_1, B_2) = 0.5 \times 0.25 = 0.125$. For the Movie data set in Section 1, then $\delta_{Director}^A(Scorsese, Coppola) = \delta_{Director}^A(Coppola, Coppola) = 0.33$, and $\delta_{Director}^A$ $(Koster,\ Coppola) = 0$ while $\delta_{Director}^A$ $(Koster, Hitchcock) = 0.25$. They correspond to the fact that "*Scorsese*" and "*Coppola*" are very similar directors just as "*Coppola*" is to himself, and the similarity between "*Koster*" and "*Hitchcock*" is larger than that between "*Koster*" and "*Coppola*", as clarified in Section 1.

After specifying *IaAVS* and *IeAVS*, a coupled similarity between objects is built based on *CAVS*. Then, we consider the sum of all these *CAVS*s analogous to the construction of Manhattan dissimilarity [**?**]. Formally, we have:

Definition 3.6 *Given an information table S, the* ***Coupled Object Similarity (COS)*** *between objects* u_{i_1} *and* u_{i_2}*:*

$$COS(u_{i_1}, u_{i_2}) = \sum_{j=1}^{n} \delta_j^A(x_{i_1 j}, x_{i_2 j}), \tag{3.8}$$

where δ_j^A *is the* CAVS *measure defined in (3.1),* $x_{i_1 j}$ *and* $x_{i_2 j}$ *are the attribute values of feature* a_j *for objects* u_{i_1} *and* u_{i_2} *respectively, and* $1 \leq i_1, i_2 \leq m$*,* $1 \leq j \leq n$*.*

For *COS*, all the *CAVS*s with each feature are summed up for two objects. For example (Table **??**), $COS(u_2, u_3) = \sum_{j=1}^{3} \delta_j(x_{2j}, x_{3j}) = 0.5 + 0.125 + 0.125 = 0.75$.

4 Coupled Similarities Based Classification

In this section, we proposed a novel classification method based on the coupled similarity metric. Given a data set $\mathcal{D} =< F, C >$, $F = \{f_1, f_2, \ldots, f_n\}$ are the features of the data set, and $C = \{c_1, c_2, \ldots, c_n\}$ are the classes of the data set. This work aims to extract more information between feature to feature and feature to class by applying the coupled similarities. In terms of the distance based classification task, the K-Nearest-Neighborhood(KNN) is the most popular method, however, it lack of efficiency when it does the classification. More precise, when the KNN algorism does the classification, it needs to compute the distance between one object to each of others object to find the K-nearest object, and then to judge wether this object is belonged to. In contrast, we proposed a method that only calculate the distance between the object to the cluster centers. Actually, this is a generalized process to find the most representative object to stand for the similar objects. The experiments showed that the proposed method reduce the time of classification substantially, and it does not loose much classification accuracy.

4.1 Clustering Within the Class

In this section, a coupled similarities based clustering method will be illustrated. By the definition 3.8, a coupled similarity can be calculated between two objects $COS(d_i, d_j) = \sum_{j=1}^{3} \delta_j(x_{2j}, x_{3j})$. For the classification task, we generating the coupled similarities within one class first because we assume that there might be more coupled relations within one class than between two classes. In order to enhance the speed of the clustering process, we enumerated all the object within the data set $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ in to a comparison table $\mathcal{T}$ and then calculated the coupled similarities between each of them. Since our definition of similarity is a relative value, it only can be applied when given two objects, which means it cannot create a middle point of two objects. Furthermore, the mean of two categorical attribute cannot be calculated as well, for instance, it is hard to say what gender is between male and female. As a result, the traditional clustering method like K-Means cannot be applied directly, because it couldn't find the mean point within a group of object. To solve this problem, we used the Spherical K-Means clustering method to instead of K-Means as our clustering method.

Table 1: Coupled Similarity Between Objects

Object Pairs	Similarity
d_1, d_2	0.23
d_1, d_3	0.31
.	.
.	.
.	.
d_n, d_m	s

Spherical K-Means Clustering

Let d_1, d_2 be two categorical object from the data set $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, the similarities among the objects is based on the definition 3.8. The clustering process is to partition the data set $\mathcal{D}$ into T clusters, each of the cluster can be named as $\mathcal{C} = \{c_1, c_2, \ldots, c_t\}$ respectively. The perfect solution can be formally describe as the following maximization problem:

$$\{c_t\}_{j=1}^{k} = \arg\max_{\{c_t\}_{t=1}^{k}} \sum_{t=1}^{k} \sum_{d_i \in c_t} Cos(m_t, d_i) \tag{4.1}$$

$\{c_t\} = \{d_{t1}, d_{t2}, \ldots, d_{tn}\}$ is a cluster with certain objects, A centroid point m_t of cluster c_t is a object within the c_t which has the minimal similarity to all other objects within the cluster, for any object d' in c_t, the centroid point m_t that

$$\sum_{d_i \in c_t} Cos(m_t, d_i) \leq \sum_{d_i \in c_t} Cos(d', d_i) \tag{4.2}$$

The clustering method is straightforward, very similar to K-Means. Firstly, it randomly chooses K object from data set $\mathcal{D}$ as the centroid object m_k, m stands for the temp mode of the cluster and k is the cluster id for each cluster. Secondly, it allocates each object d_n to theirs nearest centroid object m_k as a intermediate cluster c_k, where c_k contains a set of objects $\{d_{k1}, d_{k2}, \ldots, d_{kn}\}$ which are the nearest objects to this centroid object m_k. Thirdly, it searches for a new centroid object within each cluster c_k, the new centroid object is the object which has minimal similarity to all other object with in the cluster.When the new centroid object has been confirmed, repeat to assign each object to the new centroid object to reform the cluster. Finally, iterate the process until the centroid

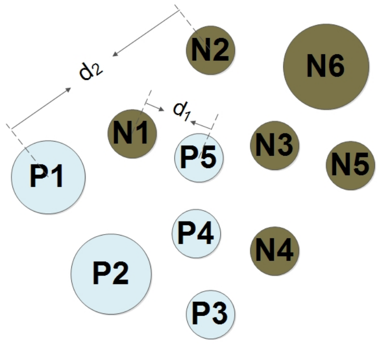

Figure 1: The training data set after clustering

object is fixed for any cluster c_t.

$$m_t^n = m_t^{n+1} \tag{4.3}$$

n stands for the iteration times. Meanwhile, in some extreme case, the centroid object cannot be fixed at all, there is an alternative criterion that

$$|(\sum_{t=1}^{k}\sum_{d_i\in c_t} Cos(m_t,d_i))^n - (\sum_{t=1}^{k}\sum_{d_i\in c_t} Cos(m_t,d_i))^{n+1}| \leq \varepsilon \tag{4.4}$$

Same as above n is the iteration times, ε is the certain threshold, if the "change" of the cluster after iteration is not significant, the searching algorithm will stop. The Spherical K-Means clustering method prevents the problems which K-Means leads to, thus it suitable for our coupled similarity based clustering.

Classification with Weighted Clusters

For the binary classification task, once the clustering process has been finished, we have several clusters within both positive and negative class. Moreover, each centroid of the cluster has its unique value for classification, due to the difference of coupled distanced between them are substantial.

As the $Figure 1$ shows, each circular stands for a cluster, and the caption of the circular P_j and N_j stand for centroid object in both the positive class and negative class respectively, moreover, we also use the different color to present the clusters which belongs to a different class. The d_1 and d_2 is the coupled similarity between two centroid P_5 and N_1 and P_1 and N_2. Apparently, the coupled similarity d_2 is significantly larger than d_1. When doing a classification task, this difference will affect the result remarkably. Unfortunately, the classic KNN algorithm neglect this and judge every point as equal significance. More precisely, when it does a classification job with an incoming object, and it only count the amount of the nearest neighbors which class belongs to, without considering the unique value of its neighbors. This work proposed a novel classifier with weighted clusters, which comprehensively involved the information of the coupled similarity from every centroid object, and by utilizing this information to make a certain weight to each centroid object, finally, classification by accumulating those weights.

5 Experiment and Evaluation

In this section, several experiments are performed on extensive UCI data sets to show the effectiveness and efficiency of our proposed coupled similarities. The experiments are divided into two categories: coupled similarity comparison and *COS* application. For simplicity, we just assign the weight vector $\alpha = (\alpha_k)_{1\times n}$ with values $\alpha(k) = 1/n$ in (3.7).

5.1 Coupled Similarity Comparison

To compare efficiencies, we conduct extensive experiments on the inter-coupled relative similarity metrics: *IRSP*, *IRSU*, *IRSJ*, and *IRSI*. The goal in this set of experiments is to show the obvious superiority of *IRSI*, compared with the most time-consuming measure *IRSP*. As discussed in Section **??**, the computational complexity linearly depends on the time costs of *ICP* with given data. Thus, we consider a comparison of complexities represented by the time costs of *ICP*. Also explained in Section **??**, the complexity for *IRSP* is $O(n^2R^2 2^R)$, while the other three have equal smaller complexity $O(n^2R^3)$. Here, scalability analysis is explored in terms of these two factors separately: the number of features $|A|$ and the maximal number of attribute values R.

From the perspective of $|A|$, Soybean-large data set is considered with 307 objects and 35 features. Here, we fix R to be 7, and focus on $|A|$ ranging from 5 to 35 with step 5. In terms of the total time costs of *ICP*, the computational complexity comparisons among four measures (*IRSP*, *IRSU*, *IRSJ*, and *IRSI*) are depicted in Figure **??**($|A|$). The result indicates that the complexities of all these measures keep increasing when $|A|$ becomes larger. The acceleration of *IRSP* (from 3328 to 74128) is the greatest compared with the slightest acceleration of *IRSI* (from 632 to 15704). Apart from these two, the scalability curves are almost the same for *IRSU* and *IRSI*, though the complexity of *IRSU* is slightly higher than that of *IRSJ* with varied $|A|$. Therefore, *IRSI* is the most stable and efficient measure to calculate the intra-coupled relative similarity in terms of $|A|$.

From the perspective of R, the variation of R is considered when $|A|$ is confirmed. Here, we take advantage of the Adult data set with 30718 objects and 13 features chosen. Specifically, the integer feature "fnlwgt" is discretized into different intervals (from 10 to 10000) to form distinct R ranging from 16 to 10000, since one of the existing categorial attributes "education" already has 16 values. The outcomes are shown in Figure **??**(R), in which the horizontal axis refers to R, and the vertical axis indicates the relative complexity ratios in terms of $\xi(J/U)$, $\xi(I/J)$, and $\xi(I/U)$. From this figure, we observe all the ratios between 10% and 100%, which again verifies the complexity order for these four measures indicated in Section **??**. Another issue is that all three curves decrease as R grows, which means the efficiency advantages of *IRSJ* upon *IRSU* (from 85.5% to 46.8%), *IRSI* upon *IRSJ* (from 78.2% to 40.2%), and *IRSI* upon *IRSU* (from 66.9% to 18.8%) all become more and more obvious with the increasing of R. The general trend of these ratios always falling comes from the fact that there is a higher probability of getting a join set smaller than the whole set, and an intersection set smaller than the join set, with larger R. The same conclusion also holds for the ratio $\xi(U/P)$, but this is due to the fact that $q^{-1}(x) = x/2^x$ is a strictly monotonously decreasing function when $x > 1$. We omit this ratio in Figure **??**(R) since the denominator $|ICP^{(P)}|$ becomes exponentially large when R grows, e.g., it equals to 5.12×10^{83} when $R = 500$. Hence, *IRSI* is the least time-consuming intra-coupled similarity with regard to R.

In summary, all the above experiment results clearly show that *IRSI* outperforms *IRSP*, *IRSU*, and *IRSJ* in terms of the computational complexity. In particular, with the increasing numbers of either features or attribute values, *IRSI* demonstrates superior efficiency compared to the others. *IRSJ* and *IRSU* follow, with *IRSP* being the most time-consuming, especially for the large-scale data set.

5.2 Application

In this part of our experiments, we focus on the computational accuracy comparison. In the following, we evaluate the *COD* which is derived from (3.8):

$$COD(u_{i_1}, u_{i_2}) = \sum_{j=1}^{n} h_1(\delta_j^{Ia}(x_{i_1j}, x_{i_2j}))\cdot h_2(\delta_j^{Ie}(x_{i_1j}, x_{i_2j})), \tag{5.1}$$

where $h_1(t)$ and $h_2(t)$ are decreasing functions. Based on intra-coupled and inter-coupled similarities, $h_1(t)$ and $h_2(t)$ can be flexibly chosen to build dissimilarity measures according to specific requirements. Here, we consider $h_1(t) = 1/t - 1$ and $h_2(t) = 1 - t$ to reflect the complementarity of similarity and dissimilarity measures. In terms of the capability on revealing the relationship between data, the better the dissimilarity induced, the better is its similarity.

To demonstrate the effectiveness of our proposed *COD* in application, we compare two clustering methods based on two dissimilarity metrics on six data sets. Here, *COD* is used with the outperforming measure *IRSI*.

One of the clustering approaches is the k-modes (*KM*) algorithm [**?**], designed to cluster categorical data sets. The main idea of *KM* is to specify the number of clusters k and then to select k initial modes, followed by allocating every object to the nearest mode. The other is a branch of graph-based clustering, i.e., spectral clustering (*SC*) [**?**], which makes use of the Laplacian Eigenmaps on dissimilarity matrix to perform dimensionality reduction for clustering prior to the k-means algorithm. In respect of feature dependency aggregations, however, Ahmad and Dey [**?**] evidenced that their proposed metric *ADD* outperforms *SMD* in terms of *KM* clustering. Thus, we aim to compare the performances of *ADD* [**?**] and *COD* (5.1) for further clustering evaluations.

We conduct four groups of experiments on the same data sets: *KM* with *ADD*, *KM* with*COD*, *SC* with *ADD*, and *SC* with*COD*. The clustering performance is evaluated by comparing the obtained cluster of each object with that provided by the data label in terms of accuracy (*AC*) and normalized mutual information (*NMI*) [**?**]. *AC*$\in [0, 1]$ is a degree of closeness between the obtained clusters and its actual data labels, while *NMI*$\in [0, 1]$ is a quantity that measures the mutual dependence of two variables: clusters and labels. *AC*$= 1$ or *NMI*$= 1$ if the clusters and labels are identical, and *AC*$= 0$ or *NMI*$= 0$ if the two sets are independent. In fact, the larger *AC* or *NMI* is, the better the clustering is, and the better the corresponding dissimilarity metric is.

Figure **??** reports the results on six data sets with different $|U|$, ranging from 15 to 699 in increasing order. In terms of *AC* and *NMI*, the evaluations are conducted with *KM-ADD*, *KM-COD*, *SC-ADD*, and *SC-COD* individually. Followed by Laplacian Eigenmaps, the subspace dimensions are determined by the number of labels in *SC*. For each data set, the average performance is computed over 100 tests for *KM* and k-means in *SC* with distinct start points.

As can be clearly seen from Figure **??**, the clustering methods with *COD*, whether *KM* or *SC*, outperform those with *ADD* in terms of both *AC* and *NMI* measures. That is to say, dissimilarity metric *COD* is better than ADD on clustering qualities. Specifically for *KM*, the *AC* improving rate ranges from 5.56% (Balloon) to 16.50% (Zoo), while the *NMI* improving rate falls within 4.76% (Soybean-s) and 37.38% (Breastcancer). With regard to *SC*, the former rate takes the minimal and maximal ratios as 4.21% (Balloon) and 20.84% (Soybean-l), respectively; however, the latter rate belongs to $[5.45\%$ (Soybean-l), 38.12% (Shuttle)$]$. Since *AC* and *NMI* evaluate clustering quality from different aspects, they generally take minimal and maximal ratios on distinct data sets. Another significant observation is that *SC* mostly outperforms *KM* a little whenever it has the same dissimilarity metric; in fact, Luxburg [**?**] has indicated that *SC* very often outperforms k-means for numerical data.

We draw the following two conclusions: 1) intra-coupled relative similarity *IRSI* is the most efficient one when compared with *IRSP*, *IRSU* and *IRSJ*, especially for large-scale data; 2) our proposed object dissimilarity metric *COD* is better than others, such as dependency aggregation only *ADD*, for categorical data in terms of clustering qualities.

6 Conclusion

We have proposed *COS*, a novel coupled object similarity metric which involves both attribute value frequency distribution (intra-coupling) and feature dependency aggregation (inter-coupling) in measuring attribute value similarity for unsupervised learning of nominal data. Theoretical analysis and substantial experiments have shown that inter-coupled relative similarity measure *IRSI* significantly outperforms the others (*IRSP*, *IRSU*, *IRSJ*) in terms of efficiency, in particular on large-scale data, while maintaining equal accuracy. Moreover, our derived dissimilarity metric is more comprehensive and accurate in capturing the clustering qualities in accordance with substantial empirical results.

We are currently applying the *COS* measure with *IRSI* to feature discretization, clustering ensemble, and other data mining tasks. We are also considering extending the notion of "coupling" for the similarity of numerical data. Moreover, the proposed concepts *Inter-information Function* and *Information Conditional Probability* for the information table have potential for other applications.

7 Acknowledgment

This work is sponsored by Australian Research Council Grants (DP1096218, DP0988016, LP100200774, LP0989721), Tianjin Research Project (10JCYBJC07500), and QCIS (Center for Quantum Computation and Intelligent Systems).

7.1 Citations

Citations within the text should include the author's last name and the year of publication, for example [Gottlob, 1992]. Append lowercase letters to the year in cases of ambiguity. Treat multiple authors as in the following examples: [Abelson *et al.*, 1985] or [Baumgartner *et al.*, 2001] (for more than two authors) and [Brachman and Schmolze, 1985] (for two authors). If the author portion of a citation is obvious, omit it, e.g., Nebel [2000]. Collapse multiple citations as follows: [Gottlob *et al.*, 2002; Levesque, 1984a].

7.2 Footnotes

Place footnotes at the bottom of the page in a 9-point font. Refer to them with superscript numbers.[2] Separate them from the text by a short line.[3] Avoid footnotes as much as possible; they interrupt the flow of the text.

8 Illustrations

Place all illustrations (figures, drawings, tables, and photographs) throughout the paper at the places where they are first discussed, rather than at the end of the paper. If placed at the bottom or top of a page, illustrations may run across both columns.

Illustrations must be rendered electronically or scanned and placed directly in your document. All illustrations should be in black and white, as color illustrations may cause problems. Line weights should be 1/2-point or thicker. Avoid screens and superimposing type on patterns as these effects may not reproduce well.

Number illustrations sequentially. Use references of the following form: Figure 1, Table 2, etc. Place illustration numbers and captions under illustrations. Leave a margin of 1/4-inch around the area covered by the illustration and caption. Use 9-point type for captions, labels, and other text in illustrations.

Acknowledgments

The preparation of these instructions and the LaTeX and BibTeX files that implement them was supported by Schlumberger Palo Alto Research, AT&T Bell Laboratories, and Morgan Kaufmann Publishers. Preparation of the Microsoft Word file was supported by IJCAI. An early version of this document was created by Shirley Jowell and Peter F. Patel-Schneider. It was subsequently modified by Jennifer Ballentine and Thomas Dean, Bernhard Nebel, and Daniel Pagenstecher. These instructions are the same as the ones for IJCAI–05, prepared by Kurt Steinkraus, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Lab.

[2] This is how your footnotes should appear.

[3] Note the line separating these footnotes from the text.

A LaTeX and Word Style Files

The LaTeX and Word style files are available on the IJCAI–13 website, `http://www.ijcai-13.org/`. These style files implement the formatting instructions in this document.

The LaTeX files are `ijcai13.sty` and `ijcai13.tex`, and the BibTeX files are `named.bst` and `ijcai13.bib`. The LaTeX style file is for version 2e of LaTeX, and the BibTeX style file is for version 0.99c of BibTeX (*not* version 0.98i). The `ijcai13.sty` file is the same as the `ijcai07.sty` file used for IJCAI–07.

The Microsoft Word style file consists of a single file, `ijcai13.doc`. This template is the same as the one used for IJCAI–07.

These Microsoft Word and LaTeX files contain the source of the present document and may serve as a formatting sample.

Further information on using these styles for the preparation of papers for IJCAI–13 can be obtained by contacting `pcchair13@ijcai.org`.

References

[Abelson *et al.*, 1985] Harold Abelson, Gerald Jay Sussman, and Julie Sussman. *Structure and Interpretation of Computer Programs*. MIT Press, Cambridge, Massachusetts, 1985.

[Baumgartner *et al.*, 2001] Robert Baumgartner, Georg Gottlob, and Sergio Flesca. Visual information extraction with Lixto. In *Proceedings of the 27th International Conference on Very Large Databases*, pages 119–128, Rome, Italy, September 2001. Morgan Kaufmann.

[Brachman and Schmolze, 1985] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April–June 1985.

[Gottlob *et al.*, 2002] Georg Gottlob, Nicola Leone, and Francesco Scarcello. Hypertree decompositions and tractable queries. *Journal of Computer and System Sciences*, 64(3):579–627, May 2002.

[Gottlob, 1992] Georg Gottlob. Complexity results for nonmonotonic logics. *Journal of Logic and Computation*, 2(3):397–425, June 1992.

[Levesque, 1984a] Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artificial Intelligence*, 23(2):155–212, July 1984.

[Levesque, 1984b] Hector J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence*, pages 198–202, Austin, Texas, August 1984. American Association for Artificial Intelligence.

[Nebel, 2000] Bernhard Nebel. On the compilability and expressive power of propositional planning formalisms. *Journal of Artificial Intelligence Research*, 12:271–315, 2000.