Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2015) 23: 1195 – 1208
© TÜBİTAK
doi:10.3906/elk-1307-127

New metrics for clustering of identical products over imperfect data

Zeki YETGİN[*], **Furkan GÖZÜKARA**
Computer Engineering Department, Mersin University, Mersin, Turkey

Received: 16.07.2013 • **Accepted/Published Online:** 08.01.2014 • **Printed:** 10.06.2015

Abstract: This paper introduces the concept of product identity-clustering based on new similarity metrics and new performance metrics for web-crawled products. Product identity-clustering is defined here as the clustering of identical products, e.g., for price comparison purposes. Products blindly crawled over web sources, e.g., online marketplaces, have different description formats, where the features describing the same products differ in both number and representation formats. This problem causes imperfect feature vectors, where the vectors are considered to be not uniform in length and structure, with the features of various data types (numeric, categorical), and unknown vector structures. Furthermore, the product information usually contains redundant, missing, or faulty data, which are regarded as noise here. Product identity-clustering becomes a challenge when the vectors' metadata are previously unknown and the imperfect nature of the feature vectors is considered with the occurrence of noise.

In this paper, the product identity-clustering concept is introduced as a new mining metric in e-commerce. Then novel similarity metrics are introduced to improve the product identity-clustering performance of legacy metrics. Finally, novel performance metrics are proposed to measure the performance of the identity-clustering algorithms. Using these metrics, a comparison of the legacy-based similarity metrics (Euclidian, cosine, etc.) and the proposed similarity metrics is given. The results show that legacy metrics are not successful in discriminating identical web-crawled products and the proposed metrics enable better achievement in the product identity-clustering problem.

Key words: Product clustering, similarity metrics, identity clustering, performance metrics, web mining

1. Introduction

With advances in web service technologies, collaboration of information-processing sources using web services is becoming more and more popular. Now, online marketplaces are becoming powerful product search engines integrated with various appealing services for their users (sellers and customers), such as recommended products [1] or product comparisons with other sellers. Particularly, online trading services covering many online marketplaces include business-to-business trade, online retail, and data-centric cloud computing services. These web platforms can collect information from many heterogeneous web sources (e.g., online markets), unify them, and present the results to the customers who initiated the query for a particular product.

One general issue in these search platforms is the so-called identity-clustering problem [2]. Usually, sellers and customers want to see the same products from different sellers in order to compare the products in terms of features such as price. Product identity-clustering requires finding the products with the same features. However, the same product is usually described differently by different sellers on the web. Most of the time, redundant words are added to the product description in order to, e.g., increase the appeal of the

[*] Correspondence: zyetgin@mersin.edu.tr

product. There are also erroneous cases where the product descriptions have missing or faulty features caused by the users or the system. In real-life decision-making problems, preferences are vague and decision-relevant information is imperfect as described in natural language [3]. Issues related to natural language such as the effects of diacritics [4] also make the decision-making process hard. Importantly, each web source has its own internal metadata (schema) to describe the structure of the product information. Recently, product information from various web sources has been enabled to be merged using ontology mapping approaches [5,6]. Ontology mapping is the schema-matching approach used so web sources can understand their metadata descriptions for product information. However, product information ontology must be developed manually for each web shopping source. It is important to note that ontology mapping does not guarantee perfect information, which means the gathered product information may still involve unstructured or incomplete attributes. In order to cope with the unstructured or imperfect information, new decision theories apart from ontology matching approaches are required in each phase of product clustering.

Formally, clustering is a technique used to group similar objects into the same categories according to a similarity measure. Each object is described by a feature vector and the similarity measures define the degree of similarity for any pair of vectors in a vector space. With product identity-clustering, any pair of products is identical if they fall into the same category. Thus, identity-clustering performance depends highly on the underlying similarity metrics, defined for the feature vectors in a problem domain. However, with an unstructured and imperfect dataset where the length and metadata of the feature vectors are not stable, the similarity between the vectors becomes hard to measure. Thus, identity-clustering requires new metrics to measure similarities between unstable feature vectors.

Clustering has many applications in various domains, including text extraction/summarization [7,8], market segmentation [9], web access pattern analysis [10], and product recommendation [11]. However, the identity-clustering cannot be applicable to all clustering domains. For example, considering market segmentation, clustering the needs and purchase behaviors of two customers that are exactly the same is hardly possible or is even impossible in practice. Similarly, looking for web access patterns that are exactly the same is impossible and meaningless. The study of identity-clustering requires its own methods to alleviate the problems specific to it.

In the literature, the term “identity-clustering” was first introduced for clustering identical persons. In [2], identity-clustering is applied to a set of human faces as a face recognition task to cluster the faces of the same people. Similarly, [12] supports the clustering of web people from the results of person queries over search engines. Product comparison using ontology mapping/matching approaches [5,6] is closely related to the product identity-clustering with respect to their aims. Ontology-based approaches are schema merging techniques that enable the product comparisons over the merged and (semi-) structured dataset whose metadata are known by the ontology. However, the product identity-clustering is a decision-making technique to identify identical products, even if the dataset is not structured and the metadata of the dataset is unknown. Thus, they are complementary, rather than competitive, methods.

Other studies related to product-clustering aim either for analysis of customer behaviors [13], e.g., to cluster recommended products of interest [1], or analysis of product reviews [14–16], e.g., to cluster the product features rather than products. The analyses of product reviews study human opinions about the product features. These studies usually use sentiment analyses [14,15] or opinion mining [16–20], where human subjects are involved in assessing the product features. For example, [20] aims to provide a summary of human opinions based on product features. It clusters the synonym features and tries to explore the correlation between the human reviews and a set of features of the products. Commonly, feature extractions of the products are also studied with respect to opinion-mining or human behaviors. In [16–19], information extraction systems are introduced, which extract fine features with respect to associated opinions. None of the studies in the literature examine clustering of identical products, e.g., for comparison purpose, over an imperfect and unstructured dataset, such as a web crawled dataset. The product identity-clustering is regarded here as new mining metric in e-commerce, not considered before in the literature.

In this paper, a crawler for a set of online e-trade systems is implemented to extract products' features in many categories. Then product identity-clustering is demonstrated using hierarchical clustering algorithms of various types. The performance test is done with the proposed performance metric for the identity-clustering problem, which is one of the contributions of the paper. Then some similarity metrics are proposed to improve the product identity-clustering performance of legacy metrics.

The paper is organized as follows. The second section provides the system model and formulization of the problem. The third section demonstrates the experimental results of the proposed methods. Finally, conclusions and future directions are given.

2. System model

2.1. Feature extraction method

In this section, the feature extraction method used for clustering algorithms is described. A simple feature extraction model is considered in the study. The words describing a particular product are called features, and a set of extracted features of particular interest are called the feature vector of the product. In order to generate feature vectors, some steps are needed beforehand, e.g., to eliminate simple noise. In our case, the following steps are executed: 1) removing single-length characters from the dataset, 2) replacing nonletter characters with space characters where only the characters "a–z" and "0–9" are allowed, and 3) lowercasing the dataset. These processes eliminate the noise at the character level. However, for further noise elimination, a more sophisticated algorithm is necessary. As an example, the raw features and the extracted features are shown in Tables 1 and 2, respectively.

Table 1. Examples of web-crawled raw data for the Samsung Netbook product group.

Product ID	Product description
9065	Samsung N150-JP0XTR N570 2GB 320GB 10.1"" W7STR
4907	SAMSUNG N150-JP0XTR BEYAZ INTEL ATOM N570 1.66 GHz-2048MB DDR3-320GB -10.1"-CAM-BT-W7STR
4875	SAMSUNG N150-JP0XTR Atom N570 1.66GHZ 2GB 320GB 10.1" Netbook W7S Beyaz
169120	Samsung N150JP0XTR N570 2G 320GB 10.1 W7S BEYAZ
168088	SAMSUNG N150-JP0XTR W

Table 2. Examples of web-crawled data after simple noise elimination.

Product ID	Product description
9065	samsung n150 jp0xtr n570 2gb 320gb 10 w7str
4907	samsung n150 jp0xtr beyaz intel atom n570 66 ghz 2048mb ddr3 320gb 10 cam bt w7str
4875	samsung n150 jp0xtr atom n570 66ghz 2gb 320gb 10 netbook w7s beyaz
169120	samsung n150jp0xtr n570 2g 320gb 10 w7s beyaz
168088	samsung n150 jp0xtr

Some features of the products may have false forms, such as product 169120 in Table 1, where the features "N150" and "JP0XTR" are in concatenated form and seem like a single feature. There may also be missing features of the products like product 168088 in Table 1, where some of the important features are absent. Therefore, before the feature extraction phase, the dataset should be cleaned and normalized against various erroneous cases. The product features collected over the web sources show that sophisticated methods are really needed to extract/select/transform features to achieve product identity-clustering. However, in this paper only feature transform is considered. Linguistic corrections and other corrections done beforehand are considered to be out of the scope of this study.

2.2. Clustering model

In this section, the clustering methods considered for product identity-clustering are described and the proposed similarity metrics to improve the performance of the identity-clustering algorithms are formulated. We have considered hierarchical types of clustering to demonstrate how the clustering algorithms perform well in clustering identical products. There are two important parameters for clustering algorithms: similarity metrics and linkage metrics. The similarity metric, also called a distance metric, decides the degree of similarity between any pair of points (vectors) in the feature space. We have considered traditional similarity metrics such as Euclidian, cosine, Jaccard, and Hamming similarities, which we call legacy similarity metrics here. The legacy metrics work better on feature vectors that are uniform in both length and structure. For nonuniform feature vectors, new metrics are required. Therefore, we have demonstrated the performance of four other similarity metrics, referred to here as minimally-normalized intersection similarity (MNI), globally-normalized locally-weighted similarity (GNLW), globally-normalized indexed similarity (GNI), and globally-normalized globally-weighted similarity (GNGW), where the MNI is legacy and the others are our new contributions to the literature. Let *Vector(i)* show the feature vector of the *ith* product among N products; then the similarity between *Vector(i)* and *Vector(j)* are calculated for the MNI as follows:

$$Similarity(i,j) = \frac{|Vector(i) \cap Vector(j)|}{\min\left(|Vector(i)|, |Vector(j)|\right)} \tag{1}$$

where *Vector(i)={feature | each feature is a descriptive word of the product i}*

Similarity (i, j) creates the matrix *Similarity*, where its elements are the similarities between the product pairs at row i and column j. The similarity measure formulated in Eq. (1) states that each matched feature has a constant importance. Instead of counting 1 for each matching, a degree of importance between 0 and 1 could be given to the matched features. As a simple attempt, the GNLW, formulated in Eqs. (2)–(6), is proposed, where each feature has a weight of importance according to both its ability to differentiate the product and its frequency in the dataset. In traditional clustering algorithms, such as for document clustering [21,22], the words with higher frequencies have more importance due to the fact that the frequently-occurring words have some semantic relation with the subject of the document. However, when features of a product are considered, the words (features) with low frequencies may have a better effect in identifying the product depending on the actual descriptive performance of the words.

Since each product is uniquely described by its features, each feature in the vector could be assumed to have an equal share of the description information. Thus, the total descriptive performance of a feature could be simply regarded as the average lengths of the vectors that contain the feature. For example, if the product is described by small number of words, these words should have high importance with respect to the product in that they carry a high load of information. Let *Freqs(f)* show the frequency of the feature identified by f in the dataset and the *SpaceSum(f)* be the sum of the feature vector lengths of the feature f, then the similarity between *Vector(i)* and *Vector(j)* for the GNLW is formulated using the sequential operations in Eqs. (2)–(6), where the similarity matrix is globally normalized to a 0–1 scale in Eqs. (5) and (6).

$$SpaceSum(f) = \sum_{i=1}^{N} \left\{ \begin{array}{lll} |Vector(i)| & , & \text{If } f \in Vector(i) \\ 0 & , & \text{Otherwise} \end{array} \right\} \tag{2}$$

$$Weight(f,i) = \frac{SpaceSum(f)/Freqs(f)}{\sum\limits_{k \in Vector(i)} SpaceSum(k)/Freqs(k)} \tag{3}$$

$$Similarity(i,j) = \sum_{k \in \{Vector(i) \cap Vector(j)\}} Weight(k,i) \tag{4}$$

$$Similarity(i,j) = Similarity(i,j) - \min\left(Similarity\right) \tag{5}$$

$$Similarity(i,j) = 1 - \frac{Similarity(i,j)}{\max\left(Similarity\right)} \tag{6}$$

where *Weight(f, i)* shows the importance of the feature f with respect to the product i, and "min" and "max" operators return the minimum and maximum element of the *Similarity* matrix, respectively.

The vector definitions given for the MNI and the GNLW do not necessarily use numerical identification for the features. However, using numeric features has some advantages during the processing of vectors and enables new methods for similarity measurements. As a simple solution, the GNI is given where all the features are sorted according to a hash function, and the global indexes are used to form the feature vectors. In this paper, we used a simple hash function (alphabetical order), which is acquired in the following way. First, all the features (words) in the dataset are alphabetically sorted into ascending order. Then each feature in the sorted list is given an increasing ID in the order of their occurrences where the IDs are hash values. Finally, the feature vectors are globally normalized (see Eqs. (8) and (9)). In this way, features are projected onto a new dimension so that any combination of features (feature vector) is better identified by manipulating the feature IDs as hash values. Now, the vector definition is *Vector(i)*={*id* | *id is the hash value of each feature in the product i*} and the similarity matrix for the GNI is generated using the following sequential formulas in Eqs. (7)–(9).

$$Similarity(i,j) = \sum_{k \in \{Vector(i) \cap Vector(j)\}} k \tag{7}$$

$$Similarity(i,j) = Similarity(i,j) - \min\left(Similarity\right) \tag{8}$$

$$Similarity(i,j) = 1 - \frac{Similarity(i,j)}{\max\left(Similarity\right)} \tag{9}$$

However, the GNI does not use any information for the nonmatching features. Looking from this aspect, in GNLW the only information for nonmatched features is encoded into the weights with the summation of the ratio, *SpaceSum* / *Freqs* given in Eq. (3). However, the summation is not well considered for the nonmatching features. In order to better use the information for the nonmatching features, the hash values (feature IDs) could be used together with the concept of weighted importance. As one improvement, a new similarity measure, GNGW is proposed in Eqs. (10)–(13), where the nonmatching features are encoded as the sum of direct feature IDs in Eq. (11). Similar to the GNLW, each feature has also a weight describing its importance, formulated in Eq. (10). The difference between the weights in GNLW and GNGW is that GNLW considers local weight with respect to a particular product, whereas GNGW uses a global weight for the feature. However, with GNGW the effect of the global weight on the similarity measure is reduced by taking their logs, which in turn increases the priority of hash values. The vector definition is same as in GNI. *Vector(i)*={*id* | *id is the hash value of each feature in the product i*} .

$$Weight(f) = \frac{1 + \log\left(SpaceSum(f)\right)}{1 + \log\left(Freqs(f)\right)} \tag{10}$$

$$Similarity(i,j) = \sum_{k \in \{Vector(i) \ \cap \ Vector(j)\}} \frac{k \Big/ \sum\limits_{f \in Vector(i)} f}{Weight(k)} \tag{11}$$

$$Similarity(i,j) = Similarity(i,j) - \min\left(Similarity\right) \tag{12}$$

$$Similarity(i,j) = 1 - \frac{Similarity(i,j)}{\max\left(Similarity\right)} \tag{13}$$

Defining suitable similarity metrics is a requirement of any type of clustering algorithm. However, hierarchical clustering algorithms use an additional linkage metric, which uses the underlying distance metrics to measure the distance between subclusters. The selection of the linkage metrics decides the behavior of the algorithms while merging the subclusters to form a bigger cluster in the hierarchy. With the following linkage metrics, seven different clustering algorithms are considered in the paper. These are single (nearest distance), complete (furthest distance), average (unweighted average distance), weighted (weighted average distance), centroid (unweighted center of mass distance), median (weighted center of mass distance), and ward (minimum variance algorithm) linkage clustering. The single method considers the smallest distance between the points in two clusters for the decision of merging, whereas the complete method considers the furthest distance between two clusters. The other methods behave similarly, as their names indicate.

2.3. Performance metrics

In this section, the performance measurement metrics for the identity-clustering algorithms are formulated. Three metrics, namely false-positive (FP), false-negative (FN), and total error (TE) are introduced to assess the performance of identity-clustering when the original cluster labels are available. The metric definitions were inspired by the false-alarms and miss-detections used in the literature [23,24] for evaluating the success of change detection algorithms in remote sensing environments. These metrics consider the number of pairs (pixels of the two images) at the same position that are classified as either false-alarm or miss-detection by comparing them with the pixels of the ground-truth image at the same position. However, in our case there are no specific positions of the product pairs and all possible combinations of pairs should be considered. The proposed metric considers a space consisting of pairs where a set of clusters of pairs that the algorithm found as identical are to be compared with the original set of clusters of pairs that were acquired beforehand. Here, the metrics are redefined as follows:

i. FN indicates the number of product pairs that are classified as different by the algorithm, although they are actually identical.

ii. FP indicates the number of product pairs that are classified as identical by the algorithm, although they are actually different.

iii. TE indicates the total number of decision errors caused by either FNs or FPs

Moreover, the connectivity-index (conn-index) [25] is used as an additional clustering validation index to assess the clustering accuracy. Conn-index reflects the degree of connectedness of a cluster without using the original cluster labels and has recently regained interest due to its success in product-property modeling applications.

Co(i) shows the actual (original) cluster label of the product $i$and *Cc(i)* indicates the cluster label that the algorithm assigned to the product i, *CSize(C)* shows the number of clusters in C, *SubCluster(C, i)* represents the i. subcluster in C, *Sx(i)* represents the$i$. subcluster in *Cx.* and *Partitions(C1, C2)* shows the set of subclusters in *C1* that are acquired by mapping of *C2* to *C1,* then *FN, FP ,TE,* and their rates are formulated in Eqs. (14)–(19) and explained in the examples in Table 3.

$$FN = \sum_{i=1}^{CSize(Co)} \left[\binom{|SubCluster(Co,i)|}{2} - \sum_{k=1}^{CSize(Partitions(Cc,So(i)))} \binom{|SubCluster(Partitions(Cc,So(i)),k)|}{2} \right] \tag{14}$$

$$FP = \sum_{i=1}^{CSize(Cc)} \left[\binom{|SubCluster(Cc,i)|}{2} - \sum_{k=1}^{CSize(Partitions(Co,Sc(i)))} \binom{|SubCluster(Partitions(Co,Sc(i)),k)|}{2} \right] \tag{15}$$

$$TotalCombinationPair(C) = \sum_{i=1}^{CSize(C)} \binom{|SubCluster(C,i)|}{2} \tag{16}$$

$$FNR = \frac{FN}{TotalCombinationPair(Co)} \tag{17}$$

$$FPR = \frac{FP}{TotalCombinationPair(Cc)} \tag{18}$$

$$TER = \frac{FN + FP}{TotalCombinationPair(Co) + TotalCombinationPair(Cc)} \tag{19}$$

where $\binom{n}{2} = 0$ is assumed for all $n < 2$ due to fact that the metrics only assumes number of product pairs.

In order to demonstrate the performance metrics, two simple examples are given in Table 3 where only FNs (Example 1) or FPs (Example 2) occur over 6 products. In Example 1, a total of *C(6, 2)* combinations of pairs are actually identical according to *Co* but the algorithm misses some of the pairs. The misses are detected by the mapping of *Co* to *Cc*, which creates two different partitions in *Cc*, labeled as partition 1 (two products with cluster label 1) and partition 2 (four products with cluster labels 2). On the other hand, the FPs are detected by the mapping of *Cc* to *Co*, which creates a single partition in *Co*. That is, there is no false alarm since all the products in *Cc* are mapped to the same cluster labels in *Co*.

Table 3. Examples of FNs (Example 1) and FPs (Example 2).

Example 1		Example 2	
Cc	Co	Cc	Co
1	1	1	1
2	1	1	2
1	1	1	1
2	1	1	2
2	1	1	3
2	1	1	3

According to the formulization, the *MDR* and *FAR* for Example 1 are computed as follows:

$$MDR = \frac{\binom{6}{2} - \left[\binom{2}{2} + \binom{4}{2}\right]}{\binom{6}{2}} \text{ and } FAR = \frac{\left[\binom{2}{2} - \binom{2}{2} + \binom{4}{2} - \binom{4}{2}\right]}{\binom{2}{2} + \binom{4}{2}} = 0$$

In Example 2, there is a total of *C(6, 2)* pairs that the algorithm found as identical according to *Cc*, but some of these pairs are actually FPs according to the *Co*. Again, the FPs are detected by the mapping of *Cc* to *Co*, which creates three partitions in *Co*, labeled as partition 1, partition 2, and partition 3. The two products in partition 1, the 2 products in partition 2, and the two products in partition 3 seem the same according to the *Cc*, although they cause many pairs of FPs. However, the mapping of *Co* to *Cc* creates a single partition in *Cc*, which states that the algorithm does not have any FNs. The *MDR* and *FAR* for Example 2 are computed as follows;

$$MDR = \frac{\left[\binom{2}{2} - \binom{2}{2} + \binom{2}{2} - \binom{2}{2} + \binom{2}{2} - \binom{2}{2}\right]}{\binom{2}{2} + \binom{2}{2} + \binom{2}{2}} = 0 \text{ and}$$

$$FAR = \frac{\binom{6}{2} - \left[\binom{2}{2} + \binom{2}{2} + \binom{2}{2}\right]}{\binom{6}{2}}$$

For demonstration purposes, Table 3 only considers a single cluster in *Co* in Example 1, and a single cluster in *Cc* in Example 2. When many sets of subclusters are considered in both *Co* and *Cc*, the same processes are repeated for each subcluster similarly. The proposed metrics automatically compute the performance of the identity-clustering algorithms.

3. Results

3.1. Datasets

One million products are crawled from the 20 most popular online market places in Turkey. Six of them are listed in Table 4 for demonstration purposes. The products have many different categories including computers, books, cosmetics, home appliance, etc. The web crawler is designed to gather product information along with other page information using the schema descriptions of each of the online sellers' search engines. However, the product information gathered is unstructured, and it is merged into a single text file where each line describes a particular product, as shown in Table 1.

Table 4. Some online shopping malls that are used to collect product information.

Web Site	Number of products crawled	Number of pages crawled
hepsiburada.com	177,310	313,946
hizlial.com	84,046	166,197
webdenal.com	69,979	121,853
ereyon.com.tr	68,960	92,076
pratikev.com	63,170	69,275
netsiparis.com	40,525	59,294

We have selected 100 types of products randomly and collected 1000 products of the selected types, again randomly from the dataset. For the 1000 products, we manually visited the web sites of their online sellers and generated the error-free class labels of each product, which is denoted by the *Co* array in the problem formalization. The algorithms produce *Cc* arrays based on the same dataset of 1000 products. The feature extraction phase (Section 2.1) is applied to the raw data to generate the input dataset.

3.2. Experimental results

In this section, we provide the results of the identity-clustering algorithms based on the legacy similarity metrics and the proposed similarity metrics. The evaluation of each clustering experiment is done by the proposed performance metrics and conn-index validation scores where the smaller score is better. The results are given in Tables 5–13.

Table 5. Identity clustering results with cosine/Euclid similarity.

Similarity	Cosine/Euclid Similarity			
Linkage	False-negative rate (FNR)	False-positive rate (FAR)	Total error rate (TER)	Conn-index
Single	0.89–0.92	0.01–0.03	**0.85**–0.90	**280**–400
Complete				
Average				
Weighted				
Centroid				
Median				
Ward				

Table 6. Identity-clustering results with Hamming similarity.

Similarity	Hamming similarity			
Linkage	False-negative rate (FNR)	False-positive rate (FAR)	Total error rate (TER)	Conn-index
Single	0.87	0.00	0.81	479.1
Complete	0.81	0.00	0.74	390.2
Average	0.79	0.00	0.72	341.6
Weighted	0.73	0.01	0.63	328.7
Centroid	0.84	0.00	0.78	387.6
Median	0.81	0.00	0.74	367.7
Ward	0.65	0.02	**0.54**	**233.8**

Table 7. Identity-clustering results with Jaccard similarity.

Similarity	Jaccard similarity			
Linkage	False-negative rate (FNR)	False-positive rate (FAR)	Total error rate (TER)	Conn-index
Single	0.58	0.42	0.52	202.9
Complete	0.65	0.07	0.54	243.9
Average	0.50	0.25	0.41	169.7
Weighted	0.50	0.07	0.37	167.4
Centroid	0.78	0.03	0.70	332.4
Median	0.63	0.07	0.52	246.3
Ward	0.51	0.09	**0.39**	**144.0**

Table 8. Identity clustering results with MNI similarity.

Similarity	Minimally normalized intersection (MNI) similarity			
Linkage	False-negative rate (FNR)	False-positive rate (FAR)	Total error rate (TER)	Conn-index
Single	0.29	0.26	**0.28**	**111.3**
Complete	0.58	0.00	0.44	340.9
Average	0.48	0.11	0.35	253.8
Weighted	0.52	0.05	0.38	296.5
Centroid	0.52	0.09	0.38	313.6
Median	0.54	0.16	0.43	334.0
Ward	0.43	0.12	0.32	266.8

Table 9. Identity-clustering results with GNLW similarity.

Similarity	Globally normalized locally weighted (GNLW) similarity			
Linkage	False-negative rate (FNR)	False-positive rate (FAR)	Total error rate (TER)	Conn-index
Single	0.26	0.09	**0.19**	**132.9**
Complete	0.57	0.00	0.43	282.9
Average	0.43	0.20	0.35	236.8
Weighted	0.46	0.13	0.34	228.6
Centroid	0.44	0.42	0.43	216.9
Median	0.40	0.40	0.40	199.9
Ward	0.44	0.09	0.32	216.2

Table 10. Identity-clustering results with GNI similarity.

Similarity	Globally normalized-indexed (GNI) similarity			
Linkage	False-negative rate (FNR)	False-positive rate (FAR)	Total error rate (TER)	Conn-index
Single	0.39	0.42	0.40	179.6
Complete	0.35	0.16	0.27	150.0
Average	0.23	0.20	**0.22**	**108.0**
Weighted	0.25	0.22	0.24	121.6
Centroid	0.70	0.06	0.59	394.6
Median	0.55	0.13	0.43	237.1
Ward	0.32	0.50	0.42	124.1

Table 11. Identity-clustering results with GNGW similarity.

Similarity	Globally normalized globally weighted (GNGW) similarity			
Linkage	False-negative rate (FNR)	False-positive rate (FAR)	Total error rate (TER)	Conn-index
Single	0.21	0.18	**0.20**	**106.6**
Complete	0.54	0.03	0.41	264.2
Average	0.40	0.24	0.33	202.8
Weighted	0.37	0.13	0.28	185.9
Centroid	0.37	0.30	0.34	172.6
Median	0.35	0.36	0.35	154.8
Ward	0.47	0.14	0.36	227.5

We observe that hierarchical clustering algorithms based on the legacy metrics have very poor performance due to nonuniform feature vectors, whereas the hierarchical clustering algorithms based on the proposed metrics produce much better results due to their robustness to nonuniform feature vectors. The total error rate (TER) for hierarchical clustering based on cosine/Euclid similarity metrics is around 85%–90%, which is not acceptable. The primary reason for the bad performance is the metrics' high dependency on the uniform structure of the feature vectors.

Cosine, Euclid and Hamming similarity metrics consider the length of the smallest vector as the dimension of the feature space. Hence, some of the features of the lengthy vector are discarded without any feature selection mechanisms. The cosine and Euclid metrics are not tolerant to nonmatched entries (errors), whereas the Hamming similarity has a bit more tolerance to errors. These metrics are too sensitive to the false-negative errors because they consider each feature vector as an arrangement (permutation) of features, rather than a combinational set. Thus, any miss in the position of features becomes a false-negative of the pairs. However, these metrics are very tolerant to false-positives, which in turn make the TERs a bit better than the false-negative rates.

The Jaccard, MNI, GNLW, GNI, GNGW similarity metrics are tolerant in some degree to both type of errors due to the fact that these metrics considers each feature vector as a combinational set rather than the permutation vector. The Jaccard and MNI are both intersection-based legacy metrics. However, they greatly reduce the TERs with respect to the permutation-based measures. The Jaccard shows its best performance (39%) with the Ward linkage metric, whereas the MNI achieves better in almost all linkage metrics with the best TER (28%) with single linkage. The Jaccard and MNI have also similar behaviors with respect to the decision errors except the single linkage.

The hierarchical clustering algorithms based on the proposed metrics provide a dramatic increase in performance. With our improvements, the TER is reduced up to 19%, which means that further improvements would be still necessary in identity-clustering over imperfect data. The results surely prove an achievement with respect to the legacy-based measures. Furthermore, the proposed metrics also provide an achievement with respect to the intersection-based methods such as MNI, a well-known metric that has never been applied to the product-clustering problem in the literature.

A hierarchical clustering algorithm with a particular linkage metric and a similarity metric defines a single clustering algorithm. Each clustering algorithm has its own characteristics and their application to a specific problem decides which one is to be chosen. For example, some systems work on categorical values like product-clustering, while some others may use continuous numeric values as raw features. Algorithms also differ with respect to their tolerance to decision errors. For example, some systems may have tolerance to false-negative errors but not false-positives. With the results given in Tables 12 and 13, we see that for each linkage metric one of the proposed methods provides the optimum TER and the optimum conn-index scores.

Table 12. Comparison of the results according to the proposed performance metrics.

Metrics	Best TER with legacy similarity metrics	Best TER with proposed similarity metrics
Single	0.28 (MNI)	0.19 (GNLW)
Complete	0.44 (MNI)	0.27 (GNI)
Average	0.35 (MNI)	0.22 (GNI)
Weighted	0.37 (Jaccard)	0.24 (GNI)
Centroid	0.38 (MNI)	0.34 (GNGW)
Median	0.43 (MNI)	0.35 (GNGW)
Ward	0.32 (MNI)	0.32 (GNLW)

Table 13. Comparison of the results according to conn-index validation scores.

Metrics	Best conn-index scores with legacy similarity metrics	Best conn-index scores with proposed similarity metrics
Single	111.3 (MNI)	106.6 (GNGW)
Complete	243.9 (Jaccard)	150.0 (GNI)
Average	169.7 (Jaccard)	108.0 (GNI)
Weighted	167.4 (Jaccard)	121.6 (GNI)
Centroid	313.6 (MNI)	172.6 (GNGW)
Median	246.3 (Jaccard)	154.8 (GNGW)
Ward	144.0 (Jaccard)	124.1 (GNI)

Generally, the GNLW and GNGW produce very similar results with respect to error types, but the GNGW produces better results on average. The GNGW also produces an error rate of 20%, which is very close to the global optimum (19%) obtained by the GNLW. The definition of the GNGW is derived from the concepts in the GNLW and GNI (see problem formulation). Now the results show that the GNGW really combines the advantages of GNLW and GNI. The conn-index validation scores also support this interpretation. Generally, the conn-index clustering validation complies with the proposed performance evaluation metrics and makes the GNI and GNGW candidate metrics for a general purpose identity-clustering problem.

4. Conclusions

The product identity-clustering problem is introduced as new mining metric in e-commerce. The identity-clustering problem is applied to blindly crawled web products. The performance of current clustering algorithms in solving the product identity clustering problem is demonstrated with the applied novel methodologies. The results show that current clustering algorithms have no success in the product identity-clustering problem. New identity-clustering algorithms are provided with novel similarity metrics to improve the success of the current clustering algorithms. The performance comparisons of the algorithms are demonstrated based on the proposed performance metrics for the identity clustering problem. With the results, the proposed metrics provided an achievement up to 19% in clustering performance. The metrics could also be used in search engines of online sellers for product-clustering and their interoperability test for product comparisons. However, the methods assume a very simple feature extraction model with no linguistic corrections. More work is needed to have an efficient feature extraction method for identity clustering over imperfect data.

References

[1] Chen LS, Hsu FH, Chen MC, Hsu YC. Developing recommender systems with the consideration of product profitability for sellers. Information Sciences 2008; 178: 1032–1048.

[2] Prince SJD, Elder JH. Bayesian identity clustering. In: Proceedings of the 2010 Canadian Conference on Computer and Robot Vision; June 2010; Ottawa, Canada: IEEE. pp. 32–39.

[3] Alieva R, Pedryczb W, Fazlollahid B, Huseynova OH, Alizadehe AV, Guirimove BG. Fuzzy logic-based generalized decision theory with imperfect information. Information Sciences 2012; 189: 18–42.

[4] Alpkoçak A, Ceylan M. Effects of diacritics on Turkish information retrieval. Turk J Elec Eng & Comp Sci 2012; 20: 787–804.

[5] Park S, Kim W, Lee S, Bang S. Product matching through ontology mapping in comparison shopping. In: Proceedings of IIWAS; 4–6 December 2006; Yogyakarta, Indonesia: ACS. pp. 39–49.

[6] Walther M, Jackel N, Schuster D, Schill A. Enabling product comparisons on unstructured information using ontology matching. Advances in Intelligent and Soft Computing 2011; 86: 183–193.

[7] Tiwari N, Garg S, Tiwari N. Document clustering using k-means, heuristic k-means and fuzzy c-means. In: Proceedings of the International Conference on Computational Intelligence and Communication Systems; 7–9 October 2011; Gwalior, India: IEEE. pp. 297–301.

[8] Biricik G, Diri B, Sönmez AC. Abstract feature extraction for text classification. Turk J Elec Eng & Comp Sci 2012; 20: 1137–1159.

[9] Ahmad A, Dey L, Halawani SM. A rule-based method for identifying the factor structure in customer satisfaction. Information Sciences 2012; 198: 118–129.

[10] Rajimol A, Raju G. Fol-mine - a more efficient method for mining web access pattern. In: Proceedings of the Advances in Computing and Communications; 22–24 July 2011; Kochi, India: SBH. pp. 253–262.

[11] Toma A, Constantinescu R, Nastase F. Recommendation system based on the clustering of frequent sets. WSEAS Transactions on Information Science and Applications 2009; 6: 715–724.

[12] Berendsen R, Kovachev B, Nastou EP, Rijke MD, Weerkamp W. Result disambiguation in web people search. In: Proceedings of the 34th European Conference on Advances in Information Retrieval; 1–5 April 2012; Barcelona, Spain: SBH. pp. 146–157.

[13] Galitsky B, Rosa JL. Concept-based learning of human behavior for customer relationship management. Information Sciences 2011; 181: 2016–2035.

[14] Gang L, Fei L. Application of a clustering method on sentiment analysis. Journal of Information Science 2012; 38: 127–139.

[15] Jin CN, Tun TT. Effectiveness of web search results for genre and sentiment classification. Journal of Information Science 2009; 35: 709–726.

[16] Hana J, Dongwook S, Joongmin C. Ferom: feature extraction and refinement for opinion mining. ETRI Journal 2011; 33: 720–730.

[17] Marcu D, Popescu A. Extracting product features and opinions from reviews. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing; 6–8 October 2005; Stroudsburg, PA, USA: ACL. pp. 339–346.

[18] Shu Z, Wenjie J, Yingju X, Yao M, Hao Y. Morpheme-based product features categorization in Chinese reviews mining. In: Proceedings of the 6th International Conference on Advanced Information Management and Service; December 2010; Seoul, South Korea: IEEE. pp. 324–329.

[19] Somprasertsri G, Lalitrojwong P. A maximum entropy model for product feature extraction in online customer reviews. In: Proceedings of IEEE Conference on Cybernetics and Intelligent Systems; 13–15 July 2008; Las Vegas, NV, USA: IEEE. pp. 575–580.

[20] Zhongwu Z, Bing L, Hua X, Peifa J. Clustering product features for opinion mining. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining; 9–12 February 2011; Hong Kong, China: ACM. pp. 347–354.

[21] Ponmuthuramalingaz P, Devi T. Effective term based text clustering algorithms. International Journal on Computer Science and Engineering 2010; 2: 1665–1673.

[22] Zheng HT, Kang BY, Kim HG. Exploiting noun phrases and semantic relationships for text document clustering. Information Sciences 2009; 179: 2249–2262.

[23] Çelik T, Yetgin Z. Change detection without difference image computation based on multiobjective cost function optimization. Turk J Elec Eng & Comp Sci 2011; 19: 941–956.

[24] Yetgin Z. Unsupervised change detection of satellite images using local gradual descent. IEEE Transactions on Geoscience and Remote Sensing 2012; 50: 1919–1929.

[25] Handl J, Knowles J, Kell DB. Sumplementary material to computational cluster validation in post-genomic data analysis. Bioinformatics 2005; 00: 1–3.