Two Probabilistic Models for Quick Dissimilarity Detection of Big Binary Data

ADNAN A. MUSTAFA
Department of Mechanical Engineering
Kuwait University
P.O. Box 5969 SAFAT
KUWAIT

Abstract: - The task of data matching arises frequently in many aspects of science. It can become a time consuming process when the data is being matched to a huge database consisting of thousands of possible candidates, and the goal is to find the best match. It can be even more time consuming if the data are big (> 100 MB). One approach to reducing the time complexity of the matching process is to reduce the search space by introducing a pre-matching stage, where very dissimilar data are quickly removed. In this paper we focus our attention to matching big binary data. In this paper we present two probabilistic models for the quick dissimilarity detection of big binary data: the Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (PMQDD) and the Inverse-equality Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (IPMQDD). Dissimilarity detection between binary vectors can be accomplished quickly by random element mapping. The detection technique is not a function of data size and hence dissimilarity detection is performed quickly. We treat binary data as binary vectors, and hence any binary data of any size and dimension is treated as a binary vector. PMQDD is based on a binary similarity distance that does not recognize data and its exact inverse as containing the same pattern and hence considers them to be different. However, in some applications a specific data and its inverse, are regarded as the same pattern, and thus should be identified as being the same; IPMQDD is able to identify such cases, as it is based on a similarity distance that does not distinguish between data and its inverse instance as being dissimilar. We present a comparative analysis between PMQDD and IPMQDD, as well as their similarity distances. We present an application of the models to a set of object models, that show the effectiveness and power of these models.

Key-Words: - Big data, binary data, binary vector, matching, size invariance, probabilistic model, dissimilarity detection, pattern recognition, model matching

Received: March 27, 2021. Revised: April 30, 2021. Accepted: May 6, 2021. Published: May 19, 2021.

1 Introduction

Data matching is a task that arises in many diverse fields such as, image retrieval, speech recognition, computer duplicate file detection and 3D model matching. The task can be extremely time consuming if the data are big, or the database to which the query data is being matched to is huge. One approach to reducing the time complexity of the task is to perform a pre-matching filtering stage where very dissimilar data can are quickly removed from the search space.

In this paper the focus is on matching big binary data, and in particular on the pre-matching stage of reducing the search space by removing dissimilar data quickly. Our approach in formulating binary data for matching is to represent it as ordered binary vectors and then proceed to matching these vectors.

Let a and b be two data sets that are to be compared, then they are represented as,

\[ a = [a_1 \ a_2 \ \ldots \ a_n]^T \] \[ b = [b_1 \ b_2 \ \ldots \ b_n]^T \]  

where \( a_i \) and \( b_i \), for \( i = 1 \ldots n \), are the elements of vectors a and b, respectively. \( n \) is the vector (data) size.

The original data can be of any dimension: one-dimensional (e.g. sound waves), two-dimensional: (e.g. images and matrices), three-dimensional (e.g. geographical data and MRI/CT-scans), or multi—dimensional data (satellite data). The geometry of the data can be of any type provided that the matched data have a unified ordered arrangement implying an existence of a one-to-one correspondence between all elements of the matched data. Regardless of the dimension of the data or its
geometry, all data are converted to an ordered binary vector, i.e. one-dimensional ordered data, conforming to the form shown in (1). Hence, matching binary data degenerates to matching one-dimensional ordered binary patterns represented as vectors.

The Probabilistic Matching Model for Binary Vectors (PMMBV) [1] showed that by randomly selecting vector elements between two binary vectors, and comparing their values, dissimilarity between them can be quickly detected without the need to compare the entire data. Furthermore, it showed that dissimilarity detection is vector-size invariant; dissimilarity between big vectors can be detected just as quickly as it can be detected for small vectors. Consequently, dissimilarity detection methods based on PMMBV are magnitudes faster than conventional methods that compare data on an element-by-element basis. PMMBV is based on the vector similarity coefficient ($\kappa^1$), a measure of the amount of similarity between the data defined as:

$$\kappa^1 = 1 - \kappa^0$$  \hspace{1cm} (2)

where $\kappa^1$ is the vector dissimilarity coefficient. However, PMMBV assumes non-equivalency of a data and its inverse, which may not be suitable for some applications where a data and its inverse are considered to be the same. As an example, in the field of scene analysis, an image and its inverse image contain the same pattern (i.e. same scene), and hence matching an image to its inverse image should produce a perfect match.

In this paper we present the Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (PMQDD), a model based on PMMBV for the quick dissimilarity detection of big binary data. Dissimilarity detection using PMQDD is data-size invariant; data size is not a factor in how fast dissimilarity is detected. The quickness of dissimilarity detection is governed by the vector similarity coefficient, described earlier. We also present in this paper the Inverse-equality Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (IPMQDD). IPMQDD is a modified model of PMQDD, based on a different similarity measure, the vector inverse-equality similarity coefficient, which does not distinguish between a data set and its inverse data set and considers them to be the same. Similar to PMQDD, IPMQDD is also data-size invariant. We provide a comparison of the two models and give an application of the models to data examples.

This paper is organized as follows: Section 2 presents a literature review of binary similarity distances and different approaches to matching big data. Section 3 discusses the two binary distances, the vector dissimilarity coefficient and the vector inverse-equality similarity coefficient, which are employed for PMQDD and IPMQDD, respectively. Section 4 presents the two probabilistic dissimilarity detection models, PMQDD and IPMQDD. A comparison between the two models is presented in section 5. Section 6 presents a discussion of tests conducted on a dataset and the results of applying PMQDD and IPMQDD. A conclusion of our work and the future direction of our work is presented in Section 7. An appendix at the end of the paper presents the mathematical proof of the derivation of the PMQDD and IPMQDD models.

2 Literature Review

Here we present a literature review on binary similarity measures and different approaches to matching big data cited in the literature.

2.1 Binary Similarity Distances

There are an abundant of similarity distances and measures for comparing binary vectors that have been developed over the last century and can be found in the literature [2] [3] [4] [5]. Here, we briefly review some of these measures and distances that are relevant to our work and are suited for matching binary data (see Table 1). As earlier stated, we treat binary data as one-dimensional binary vectors in our analysis. One of the earliest similarity measures developed at the beginning of the previous century is the Jaccard coefficient which measures the similarity of two vectors based on their asymmetric binary attributes [6]. The Jaccard coefficient is calculated as the ratio of the number of matches present between two binary vectors to the total size of the vector. Furthermore, the Jaccard distance is defined as the Jaccard coefficient subtracted from unity; a value of zero indicates complete similarity between two binary vectors while a value of unity indicates the absence of similarity between the vectors. The Sokal–Michener coefficient [7], also known as the Simple Matching Coefficient (SMC), is defined as the ratio of the sum of the number of matches present and the number of matches missing, to the vector size. The Simple Matching Distance (SMD) is then defined as $1 - SMC$; $SMD = 0$ indicates complete similarity between two vectors while a value of $SMD = 1$ indicates complete dissimilarity between two vectors. The Hamming distance, another binary distance introduced in the middle of the last century [8], has found applications in many fields. For binary data, the Hamming distance counts the
number of mismatches between two vectors, and hence is vector size-dependent. However, if the binary Hamming distance is normalized with respect to vector size then it becomes the Sokal–Michener coefficient. The Cosine similarity [9] is a popular similarity measure, particularly efficient for sparse vectors. It has a range from −1 to 1, where a value of −1 indicates exact inverse similarity and a value of 1 indicates exact similarity. A value of 0 indicates independence. Pearson's correlation coefficient [10] measures the linear relationship between the attributes of two vectors. With a range of [−1, 1], a value of −1 implies perfect inverse linear relationship, a value of 1 means perfect linear relationship and a value of 0 indicates linear independence. Mutual information (MI) [11] measures the mutual dependence between two variables, can also be used for matching purposes. It has found many applications in the medical field. Many MI distances have been proposed and employed, such as the variation of information [12]. The Gamma binary similarity distance (γ) [13] is a probabilistic distance that is a modification of the Hamming distance that enables similarity to be more accurately measured than employing traditional binary distances. A value of a value of γ = 0 indicates complete dissimilarity while a value of γ = 1 indicates exact similarity.

2.1 Matching Big Data
With the vast amount of data being generated today, an efficient way of analysing the data has become a critical issue. ‘Big data’ research has become an increasing topic of research interest worldwide. The field has developed immensely over the last 15 years covering many topics with a large scope of applications in diverse fields using different techniques [14] [15] [16] [17] [18] [19]. However, matching the big data in a timely manner is still an unsolved problem. Many techniques and approaches have been developed and proposed: a Markov model is suggested for city-based transport models in [20], a tensor-based framework for heterogeneous networks is described in [21], a variable neighbourhood search queue architecture is proposed for ehealth networks in [22], graph matching for large graphs common in big data is described in [23], an accelerator based on optical network-on-chip technology is employed to speed up matching in [24], and a formalization of scale-independent data and partial data matching for big data is presented in [25]. Two important concepts: feature extraction and an appropriate distance metric for improved algorithm performance is presented in [26], a deep network analyzer (DNA) is employed for big data in [27], a picture retrieval system using big data mining technology through three steps of data segmentation, mining and merging is presented in [28], a fuzzy c-means algorithm for very large data is presented in [29], an image-size invariant probabilistic model for quick dissimilarity detection for big images is presented in [30], and a survey on improved hash methods for indexing big data can be found in [31].

3 Two Binary Similarity Distances for Big Binary Data
In this section we discuss two binary similarity distances for dissimilarity detection of big binary data; the vector similarity coefficient and the vector inverse-equality similarity coefficient. The first distance

2.1 The vector similarity coefficient
From [1], the vector similarity coefficient (κ₀), is defined as a metric for measuring the similarity between two binary vectors u and v, and is defined as,

$$κ₀(u, v) = P_o((z = u ⊕ v) = 0)$$

(3)

where ⊕ is the exclusive-or operation and Pₒ denotes the probability mass function of z. As a result, κ₀∈[0,1]. Alternatively, κ₀ can be defined as,

$$κ₀(u, v) = 1 − κ(u, v)$$

(4)

where κ₁ is the vector dissimilarity coefficient which is equivalent to the Sokal-Michener metric [7]. By definition it can be seen that,

- Vectors with κ₀ = 0 implies inverse the vectors are inversely similar to each other.
- Vectors with 0 < κ₀ < 1 implies quasi-similar vectors.
- Vectors with κ₀ = 1 implies the vectors are exactly similar.

Note that Quasi-similar vectors are vectors that are neither similar nor inverse, but are in between. Thus they are vectors that have some similarity between them, even though in some cases this similarity might be minute.

2.2 The vector inverse-equality similarity coefficient
Let the vector inverse-equality similarity coefficient (δ) be defined as a quantitative measure of the amount of closeness between two binary vectors based on an element-to-element mapping. It differs from (κ₀) in that it does not distinguish between a
vector and its inverse and hence considers them to be identical. It is defined as,
\[
\delta(v_1, v_2) = |1 - 2P_{\delta}(z = v_1 \oplus v_2) = Z|, \quad Z \in \{0,1\}
\]
(5)

Hence, it is equivalent to the Gamma binary similarity distance ($\gamma$) [13]. As a result,
- Vectors with $\delta = 0$ imply that the vectors are distinct dissimilar; difference is maximized between the vectors.
- Vectors with $0 < \delta < 1$ imply that the vectors are quasi-similar vectors.
- Vectors with $\delta = 1$ imply that the vectors are similar vectors. The vectors can be either exactly the same, element-by-element, or exactly the inverse, element-by-element.

3 Two Probabilistic Models for Quick Dissimilarity Detection of Big Binary Data

In this section we present two probabilistic matching models for quick dissimilarity detection of big binary data: the Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (PMQDD) and the Inverse-equality Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (IPMQDD).

3.1 The Probabilistic Model for Quick Dissimilarity Detection of Binary vectors

The Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (PMQDD) is a model that can be used to quickly detect dissimilar binary vectors. The model is equivalent to the Probabilistic Model for Binary Vectors (PMMBV) [1]. Let $P_{QDD}$ denote the probability of detecting dissimilarity between two binary vectors, $v_1$ and $v_2$, using PMQDD. Then we can state PMQDD as follows: by randomly mapping corresponding elements of two binary vectors, the probability of detecting dissimilarity between the two vectors ($P_{QDD}$) based on the number of mappings ($p$) and the amount of similarity ($\kappa^0$) between the vectors is given by,

\[
P_{QDD}(v_1, v_2; p, \kappa^0) = 1 - (\kappa^0(v_1, v_2))^p
\]

$\kappa^0 \in [0,1], p=1,2,\ldots$ (6)

The proof is given in A1. For compactness, we will write this as,
\[
P_{QDD}(p, \kappa^0) = 1 - (\kappa^0)^p \quad \kappa^0 \in [0,1], p=1,2,\ldots
\]

A Plot of $P_{QDD}$ as a function of $p$ for several values of $\kappa^0$ is shown in Fig. 1. From the figure, it can be seen that the probability of detecting dissimilarity between two vectors quickly approaches unity after a few mappings $p$. As $\kappa^0$ increases more mappings are required to reach unity. The mean number of mappings required to detect dissimilarity between the two binary vectors, $E[p(\kappa^0)]$, is,

\[
E[p(\kappa^0)] = 1 / (1 - \kappa^0) \quad \kappa^0 \in [0,1]
\]

Fig. 2 shows a plot of the mean number of mappings required to detect dissimilarity. It can be seen that $E[p(\kappa^0)]$ is fairly constant up to $\kappa^0 = 0.7$, but then starts to increase at a higher rate as $\kappa^0$ approaches unity. Nevertheless, even at high similarity values of $\kappa^0$ only a few mappings are required to detect dissimilarity –regardless of vector size; e.g. $E[p(\kappa^0 = 0.9)] = 10$ mappings.

3.2 The Inverse-equality Probabilistic Model for Quick Dissimilarity Detection of Binary vectors

The Inverse-equality Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (IPMQDD) is a model that can be used to quickly detect dissimilar vectors. The model differs from PMQDD as it is based on a different similarity measure, the vector inverse-equality similarity coefficient, which does not distinguish between a vector and its inverse vector and considers them to be the same. Hence, IPMQDD also assumes a vector and its inverse vector to be the same when detecting dissimilarity. IPMQDD states that by randomly selecting corresponding vector elements between two binary vectors, $v_1$ and $v_2$, the probability of detecting dissimilarity between the two vectors, $P_{QDD}$, based on the number of mappings ($p$) and the amount of similarity ($\delta$) is given by,

\[
E[p(\kappa^0)] = 1 / (1 - \kappa^0) \quad \kappa^0 \in [0,1]
\]

Fig. 1. A plot of $P_{QDD}(p, \kappa^0)$ for several values of $\kappa^0$. 
Pr_{IQDD}(p, \delta) = 1 - \left( \frac{1}{2} (1 + \delta) \right)^p \left( 1 + \left( \frac{1 - \delta}{1 + \delta} \right)^p \right) \quad 0 \leq \delta \leq 1, \ p = 1, 2, \ldots \quad (9)

This can be restated as,

Pr_{IQDD}(p, \delta) = 1 - \left( \frac{1}{2} \right)^p \cdot \left( (1+\delta)^p + (1-\delta)^p \right) \quad (10)

The proof is given in A.2. A Plot of Pr_{IQDD} as a function of \( p \) for several values of \( \delta \) is shown in Fig. 3. From the figure, it can be seen that the probability of detecting dissimilarity between two vectors quickly approaches unity after a few mappings \( p \). As \( \delta \) increases more mappings are required to reach unity. The mean number of mappings required to detect dissimilarity between the two vectors for a given value of \( \delta \) is the expected value of \( p \), \( E[p(\delta)] \), and is given by,

\[ E[p(\delta)] = \frac{4}{(1 - \delta^2)^2} - 1 \quad 0 \leq \delta < 1 \quad (11) \]

Fig. 4 shows a plot of the mean number of mappings required to detect dissimilarity, \( E[p(\delta)] \). It can be seen that \( E[p(\delta)] \) is fairly constant up to \( \delta = 0.7 \), but then starts to increase at a higher rate as \( \kappa^0 \) approaches unity. Similar to \( E[p(\kappa^0)] \), even at high similarity values of \( \delta \) only a few mappings are required to detect dissimilarity –regardless of vector size; e.g. \( E[p(\delta = 0.9)] = 20 \) mappings.

4 A Comparison between PMQDD and IPMQDD

Because PMQDD and IPMQDD depend on \( \kappa^0 \) and \( \delta \), respectively, there are major differences between the performance of the two models. A summary of the major differences between PMQDD and IPMQDD appear in Table 1. As a consequence to the fact that PMQDD does not acknowledge that a vector and its inverse as being the same,

- For a given similarity value, \( Pr_{IQDD} \) approaches unity faster than \( Pr_{IPMQDD} \).
- The mean number of mappings required to detect dissimilarity between dissimilar vectors using PMQDD is less than IPMQDD. In fact, it is possible to detect dissimilarity with one mapping using PMQDD, whereas it is impossible to detect dissimilarity with one mapping using IPMQDD; a minimum of two mappings are required to detect dissimilarity between dissimilar vectors.
In this section we discuss the application of the probabilistic models PMQDD and IPMQDD to a set of 3D binary objects. PTC Mathcad 15© was used to create a set of binary objects. The set consists of the 10 objects shown in Fig. 5 (front view). The objects are virtually fitted in a cube with dimensions of 10 objects shown in Fig. 5 (front view). The entire cube containing the object and empty space is referred to as a model. Hence, each model is represented by 1x10⁶ binary voxels, If “slices” along the depth of the model are taken at each unit, then the resulting slice shows where object material exists in the slice. This is analogues to what is produced in medical MRI and CT scans. Fig. 6 shows the sliced views of model 1, where 100 slices are taken of the model, and the resulting slice is shown as a 2D picture. At each slice location, the data of the slice is read in order and saved a binary vector. By preserving the same order in recording the binary data of each model, a binary vector representation of the model is created. As a result each model is represented by a 1-mega binary vector. Figures 7, 8, 9 and 10 show the sliced views of models 3, 6, 7 and 8, respectively. In general, the following statements summarizes the set:

- All objects fill the model by volume in the range [77.3%, 77.5%].

Pairing the models with every other model in the set produces 45 different model pairs; values of $\kappa^0$ for the set are in the range of $0 – 0.999$, while values of $\delta$ for the set are in the range of $0.677 – 1.0$. Dissimilarity detection for each model pair was conducted as follows: 1000 dissimilarity detection trials were repeated and the average number of mappings found to detect dissimilarity was recorded (referred to as MDN). The maximum number of mappings attempted had a limit of $L = 3000$ mappings. This value of $L$ ensures that models with similarity of up to $\kappa^0 = 0.9996$ and $\delta = 0.9993$ can be detected. If the limit $L$ is reached for a given model pair then they are assumed to be similar. Dissimilarity detection trials were performed twice;

1. Once assuming non-equivalency of a model and its inverse model; hence the PMQDD probabilistic model was applied. The mapping detection for this set is denoted by MDN$_\kappa$.

2. Once assuming equivalency of a model and its inverse; hence the IPMQDD probabilistic model was applied. The mapping detection for this set is denoted by MDN$_\delta$.

Table 2 shows the results of dissimilarity detection using the two probabilistic models. For each model pair, $\kappa^0$, $E(\kappa^0)$ and MDN$_\kappa$ for PMQDD is displayed. This is followed by $\delta$, $E(\delta)$ and MDN$_\delta$ for IPMQDD. The values of $E(\kappa^0)$ and $E(\delta)$, obtained by (8) and (11), respectively, are tabulated for comparison purposes. As an example the first model pair consisting of models (1,2) have $\kappa^0 = 0.839$. The expected number of mappings required to detect dissimilarity –as predicted by PMQDD– is $E(\kappa^0) = 6.2$ and the actual mean number of mappings found to detect dissimilarity (with 1000 dissimilarity detection trials) is MDN$_\kappa = 5.9$. The model pair’s similarity value based on $\delta$ is $\delta = 0.679$, the expected number of mappings to detect dissimilarity –as predicted by IPMQDD– is $E(\delta) = 6.4$, and the actual mean number of mappings to detect dissimilarity (with 1000 dissimilarity detection trials) is MDN$_\delta = 6.3$. The resulting error between the predicted number of mappings $E(\kappa^0)$ and actual number of mappings MDN$_\kappa$ for this model is 5.4%. The error between the predicted number of mappings $E(\delta)$ and actual number of mappings MDN$_\delta$ for this model is 1.5%.

### Table 1: Summary of differences between IPMQDD and PMQDD.

<table>
<thead>
<tr>
<th>Item</th>
<th>Probabilistic Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability Model</td>
<td>PMQDD</td>
</tr>
<tr>
<td></td>
<td>IPMQDD</td>
</tr>
<tr>
<td>Similarity Distance</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>vector similarity</td>
</tr>
<tr>
<td></td>
<td>coefficient</td>
</tr>
<tr>
<td>Symbol</td>
<td>$\kappa^0$</td>
</tr>
<tr>
<td>Definition</td>
<td>$P,(z = 0)$</td>
</tr>
<tr>
<td></td>
<td>$</td>
</tr>
<tr>
<td>Data and inverse</td>
<td>Not Equivalent</td>
</tr>
<tr>
<td>Probability Model Equation</td>
<td>$1 - (\kappa^0)^p$</td>
</tr>
<tr>
<td></td>
<td>$1 - (\frac{1}{2})^p((1 + \delta)^p + (1 - \delta)^p)$</td>
</tr>
<tr>
<td>Expected number</td>
<td>$1 / (1 - \kappa^0)$</td>
</tr>
<tr>
<td></td>
<td>$[4 / (1 - \delta^2)] - 1$</td>
</tr>
</tbody>
</table>

* $z = v_1 \oplus v_2$, where $v_1$ and $v_2$ are the two vectors being matched.
Fig. 5. The 10 models used for testing. From left to right and top to bottom: Models ‘1’ to ‘10’.

Fig. 6. The 100 slices of model 1.

Fig. 7. The 100 slices of model 3.

Fig. 8. The 100 slices of model 6.

Fig. 9. The 100 slices of model 7.
Table 2: Dissimilarity detection results for the Model Set.

<table>
<thead>
<tr>
<th>Model pairs</th>
<th>PMODD</th>
<th>IPMODD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\kappa^2)</td>
<td>(E(\kappa^2))</td>
</tr>
<tr>
<td>1</td>
<td>0.839</td>
<td>6.2</td>
</tr>
<tr>
<td>1</td>
<td>0.998</td>
<td>634</td>
</tr>
<tr>
<td>1</td>
<td>0.840</td>
<td>6.2</td>
</tr>
<tr>
<td>1</td>
<td>0.952</td>
<td>953</td>
</tr>
<tr>
<td>1</td>
<td>0.997</td>
<td>312</td>
</tr>
<tr>
<td>1</td>
<td>0.000</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>0.161</td>
<td>1.2</td>
</tr>
<tr>
<td>1</td>
<td>0.002</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>0.840</td>
<td>6.2</td>
</tr>
<tr>
<td>2</td>
<td>0.999</td>
<td>1012</td>
</tr>
<tr>
<td>2</td>
<td>0.840</td>
<td>6.2</td>
</tr>
<tr>
<td>2</td>
<td>0.838</td>
<td>6.2</td>
</tr>
<tr>
<td>2</td>
<td>0.161</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>0.000</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>2</td>
<td>0.001</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>0.840</td>
<td>6.2</td>
</tr>
<tr>
<td>3</td>
<td>0.997</td>
<td>381</td>
</tr>
<tr>
<td>3</td>
<td>0.995</td>
<td>209</td>
</tr>
<tr>
<td>3</td>
<td>0.002</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>0.000</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>4</td>
<td>0.839</td>
<td>6.2</td>
</tr>
<tr>
<td>4</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>4</td>
<td>0.001</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>4</td>
<td>0.000</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>0.996</td>
<td>235</td>
</tr>
<tr>
<td>5</td>
<td>0.001</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>5</td>
<td>0.003</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>0.160</td>
<td>1.2</td>
</tr>
<tr>
<td>6</td>
<td>0.003</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>0.162</td>
<td>1.2</td>
</tr>
<tr>
<td>6</td>
<td>0.005</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>0.161</td>
<td>1.2</td>
</tr>
<tr>
<td>7</td>
<td>0.839</td>
<td>6.2</td>
</tr>
<tr>
<td>7</td>
<td>0.998</td>
<td>634</td>
</tr>
<tr>
<td>7</td>
<td>0.840</td>
<td>6.2</td>
</tr>
<tr>
<td>8</td>
<td>0.840</td>
<td>6.2</td>
</tr>
<tr>
<td>8</td>
<td>0.999</td>
<td>1012</td>
</tr>
<tr>
<td>9</td>
<td>0.840</td>
<td>6.2</td>
</tr>
</tbody>
</table>

* A value of \(L\) for MDN\(\kappa\) indicates that the number of mappings attempted to detect dissimilarity reached the limit without detecting dissimilarity.

Fig. 10. The 100 slices of model 8.

Fig. 11. A plot of % mapping error between \(E(\delta)\) an MDN\(\delta\) vs. \(\delta\) for model pairs with \(\delta > 0.99\).

An overall summary of results:

- The error between \(E(\kappa^2)\) and MDN\(\kappa\) was in the range \([0,6.5\%]\) with a mean error set value of 1.9% and standard deviation of 1.8%.
- The error between \(E(\delta)\) and MDN\(\delta\) was in the range \([0,14.9\%]\) with a mean error set value of 3.2% and standard deviation of 3.1%. All model pair errors were less than 10% except for the two model pairs with a high similarity value of \(\delta = 0.998\) (model pairs (1,5) and (2,4)). In fact as \(\delta\) between the models increases so does the error. This can be seen from Fig. 11 which shows a plot of MDN\(\delta\) vs. \(\delta\) for model pairs with \(\delta > 0.99\). As \(\delta\) approaches 1 the error increases greatly. This is expected due to the sensitivity of mappings with high \(\delta\).
Based on equivalency between models and their inverse, four model pairs were correctly assumed to be similar because dissimilarity between the model pairs were not detected and the maximum number of mappings attempted reached its limit of \( L = 3000 \) mappings. These were for the four instances of a model and its inverse: model pairs (1,7), (2,8), (3,9), (4,10). This is in complete agreement with \( IPMQDD \). Based on non-equivalency between models and their inverse, the same four model pairs were correctly assumed to be different from the first mapping in complete agreement with \( PMQDD \) (MDNc = 1 for all four models).

6 Conclusion

In this paper we have presented two probabilistic models, the Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (PMQDD) and the Inverse-equality Probabilistic Model for Quick Dissimilarity Detection of Binary vectors (IPMQDD) that can be used to detect dissimilarity between big binary data quickly. We model the binary data as an ordered binary vector. Both models, \( PMQDD \) and \( IPMQDD \), detect dissimilarity quickly by mapping the elements of the binary vectors. They are vector size (and hence data size) invariant; only a few mapping between the elements of the vectors are required. \( PMQDD \) is based on the vector dissimilarity coefficient (\( \kappa^0 \)), while \( IPMQDD \) is based on the vector inverse-equality similarity coefficient (\( \delta \)). \( \delta \) differs from \( \kappa^0 \) in that it does not distinguish between data and its inverse as being dissimilar. As a result, \( IPMQDD \) differs from \( PMQDD \) in that it can be employed to detect similar patterns in data, such as inverse data. Tests were conducted on a set consisting of different object models. Test results showed that both models produce similar dissimilarity detection results when the data are not an inverse of each other. When the data are an inverse, very different results are produced by the two probabilistic models in agreement with theory presented; \( PMQDD \) does not detect dissimilarity, whereas \( PMQDD \) detects dissimilarity from the first mapping. All numerical results were in close agreement with the theoretical equations presented. The mean error set was 1.9% for \( PMQDD \), and 3.2% for \( IPMQDD \). Our future research will concentrate on relaxing dissimilarity detection criteria, implying a fuzzy-like dissimilarity detection.

References:


Appendix

A.1 Proof of Pr_{QDD} Equation
Let \( d \) denote the event of occurrence of dissimilar vectors and \( s \) denote the event of occurrence of similar vectors. Let us define \( k \) to be a random variable representing the probability of event \( d \) occurring at any given mapping, such that \( 0 \leq k \leq 1 \). On the first mapping two possibilities exist; \( d \) or \( s \); the probability of occurrence of the former is \( k \) and of the latter \((1 - k)\). This is shown in Fig. XX. On the second mapping, four possibilities exist, they are \( dd, ds, sd \) and \( ss \). Their probabilities are \( k^2, k(1 - k), (1 - k)^2 \), respectively. On the third mapping 8 possibilities exist, and so on for additional mappings. It can be seen that the probability distribution of \( d \) is a Binomial distribution \([\cdot]\) given by,

\[
\varphi(X = x, p, k) = \binom{p}{x} k^x (1 - k)^{p-x}
\]

where \( X \) is a random variable denoting the number of times \( d \) occurs in \( p \) mappings and \( \varphi \) is the probability mass function of \( d \) occurring \( x \) times in \( p \) mappings. Let \( S \) denote the \( s \) events only set, \( I \) the \( d \) events only set, and \( M \) the mixed events set, defined as follows:

\[
S = \{s, ss, sss, ssss, \ldots\}
\]

\[
I = \{d, dd, ddd, dddd, \ldots\}
\]

\[
M = \{sd, sdd, sss, sds, \ldots\}
\]

The three sets: \( S, I \) and \( M \), partition the sample space. The probability of occurrence of \( S \) in \( p \) mappings, \( \Pr(S, p, k) \), is then,

\[
\Pr(S, p, k) = \varphi(X=0, p, k) = (1 - k)^p
\]

Since \( D = I \cup M = S \), then the probability of occurrence of \( D \) in \( p \) mappings, denoted by \( \Pr(D, p, k) \), is then,

\[
\Pr(D, p, k) = \varphi(0 \leq X < p, p, k) = 1 - \varphi(X = p, p, k)
\]

Hence,

\[
\Pr(D, p, k) = 1 - (1 - k)^p
\]

But by its definition we see that,

\[
(1 - k) = \kappa(u, v) = \varphi((z = (u \oplus v)) = 0)
\]

Furthermore, \( \Pr(D, p, k) \) by definition is equivalent to \( \Pr_{QDD}(p, \kappa^0) \). Then (18) produces (7),

\[
\Pr_{QDD}(p, \kappa^0) = 1 - (\kappa^0)^p
\]

\( p = 1, 2, \ldots \) and \( 0 \leq \kappa^0 \leq 1 \)

A.2 Proof of Pr_{IQDD} Equation
Since we are interested in the probability of occurrence of dissimilar vectors, then from (12),

\[
\Pr(k, p) = \varphi(0 < X < p, p, k)
\]

\[
= \sum_{x=1}^{p-1} \binom{p}{x} k^x (1 - k)^{p-x}
\]

represents the probability of occurrence of mixed events. This can be rewritten as,

\[
\Pr(k, p) = 1 - \varphi(X = 0, p, k) + \varphi(X = p, p, k)
\]

which simplifies to,

\[
\Pr(k, p) = 1 - ((1 - k)^p + k^p)
\]

But,

\[
\kappa^0 = (\delta + 1) / 2
\]

Hence using this result and (19), and substituting in (22) produces (10),

\[
\Pr_{QDD}(p, \delta) = 1 - (\frac{1}{2})^p \cdot ((1 + \delta)^p + (1 - \delta)^p)
\]