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: February 1, 2021. Revised: May 2, 2021. Accepted: May 17, 2021. Published: June 2, 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,

$$\mathbf{a} = [a_1\ a_2\ \dots\ a_n]^T \text{ and } \mathbf{b} = [b_1\ b_2\ \dots\ b_n]^T \qquad (1)$$

where a_i and b_i, for $i = 1 \dots n$, 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* (κ^0), a measure of the amount of similarity between the data defined as:

$$\kappa^0 = 1 - \kappa^1 \tag{2}$$

where κ^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 $\gamma = 0$ indicates complete dissimilarity while a value of $\gamma = 1$ 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* (κ^0), is defined as a metric for measuring the similarity between two binary vectors **u** and **v**, and is defined as,

$$\kappa^0(\mathbf{u}, \mathbf{v}) = P_o((\mathbf{z} = \mathbf{u} \oplus \mathbf{v}) = 0) \quad (3)$$

where $\oplus$ is the *exclusive-or* operation and P_o denotes the probability mass function of **z**. As a result, $\kappa^0 \in [0,1]$. Alternatively, κ^0 can be defined as,

$$\kappa^0(\mathbf{u}, \mathbf{v}) = 1 - \kappa^1(\mathbf{u}, \mathbf{v}) \quad (4)$$

where κ^1 is the *vector dissimilarity coefficient* which is equivalent to the Sokal-Michener metric [7]. By definition it can be seen that,

- Vectors with $\kappa^0 = 0$ implies inverse the vectors are inverse of each other.
- Vectors with $0 < \kappa^0 < 1$ implies quasi-similar vectors.
- Vectors with $\kappa^0 = 1$ 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 (κ^0) in that it does not distinguish between a

vector and its inverse and hence considers them to be identical. It is defined as,

$$\delta(\mathbf{v}_1, \mathbf{v}_2) = |1 - 2P_o((\mathbf{z} = \mathbf{v}_1 \oplus \mathbf{v}_2) = Z)|, \quad Z \in \{0,1\} \tag{5}$$

Hence, it is equivalent to the *Gamma binary similarity distance* (γ) [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 $\Pr_{QDD}$ denote the probability of detecting dissimilarity between two binary vectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, 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 ($\Pr_{QDD}$) based on the number of mappings (p) and the amount of similarity (κ^0) between the vectors is given by,

$$\Pr_{QDD}(\mathbf{v}_1,\mathbf{v}_2; p, \kappa^0) = 1 - (\kappa^0(\mathbf{v}_1, \mathbf{v}_2))^p$$
$$\kappa^0 \in [0,1], p=1,2,\ldots \tag{6}$$

The proof is given in A1. For compactness, we will write this as,

$$\Pr_{QDD}(p, \kappa^0) = 1 - (\kappa^0)^p \quad \kappa^0 \in [0,1], p=1,2,\ldots \tag{7}$$

A Plot of $\Pr_{QDD}$ as a function of p for several values of κ^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 κ^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,

$$E[p(\kappa^0)] = 1 / (1 - \kappa^0) \qquad \kappa^0 \in \{0,1\} \tag{8}$$

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 κ^0 approaches unity. Nevertheless, even at high similarity values of κ^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, $\mathbf{v}_1$ and $\mathbf{v}_2$, the probability of detecting dissimilarity between the two vectors, $\Pr_{IQDD}$, based on the number of mappings (p) and the amount of similarity (δ) is given by,

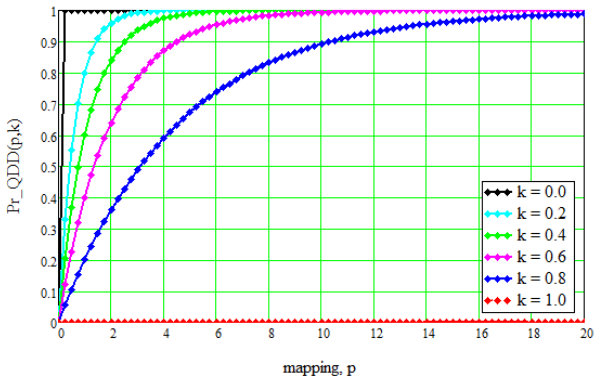

Fig. 1. A plot of $\Pr_{QDD}(p,\kappa^0)$ for several values of κ^0.

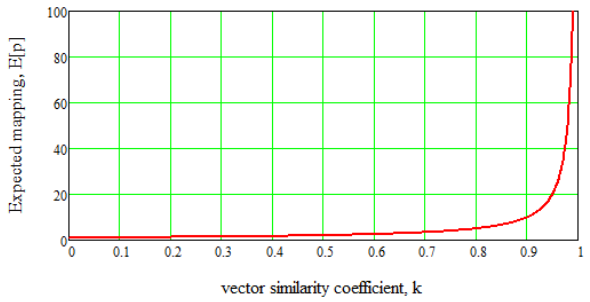

Fig. 2. A plot of $E[p(\kappa^0)]$.

$$\Pr_{IQDD}(p,\delta) = 1 - \left(\frac{1}{2}(1+\delta)\right)^p \left(1 + \left(\frac{1-\delta}{1+\delta}\right)^p\right)$$
$$0 \le \delta \le 1, p = 1, 2, \ldots \quad (9)$$

This can be restated as,

$$\Pr_{IQDD}(p,\delta) = 1 - (½)^p \cdot ((1+\delta)^p + (1-\delta)^p) \quad (10)$$

The proof is given in A.2. A Plot of $\Pr_{IQDD}$ as a function of p for several values of δ 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 δ 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 δ is the expected value of p, $E[p(\delta)]$, and is given by,

$$E[p(\delta)] = 4 / (1 - \delta^2) - 1 \qquad 0 \le \delta < 1 \quad (11)$$

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 κ^0 approaches unity. Similar to $E[p(\kappa^0)]$, even at high similarity values of δ only a few mappings are required to detect dissimilarity –regardless of vector size; e.g. $E[p(\delta = 0.9)] = 20$ mappings.

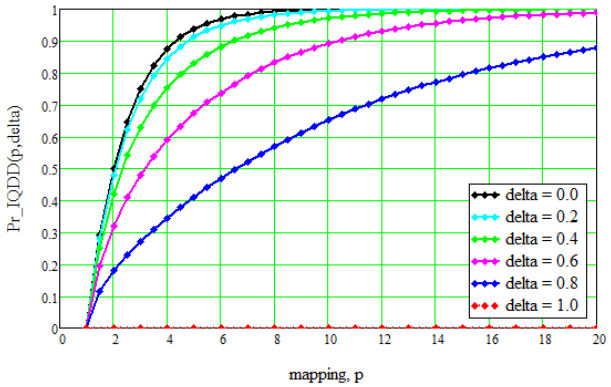

Fig. 3. A plot of $\Pr_{IQDD}(p,\delta)$ for several values of δ.

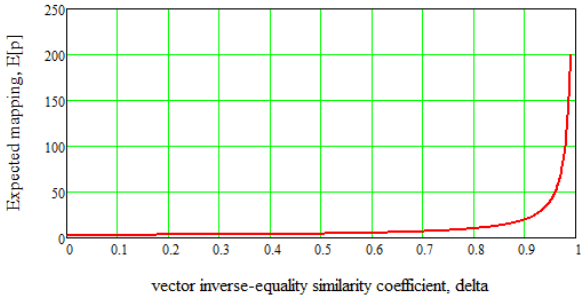

Fig. 4. A plot of $E[p(\delta)]$.

4 A Comparison between PMQDD and IPMQDD

Because *PMQDD* and *IPMQDD* depend on κ^0 and δ, 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, $\Pr_{QDD}$ approaches unity faster than $\Pr_{IQDD}$.
- 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.

Table 1: Summary of differences between *IPMQDD* and *IPMQDD*.

Item		*Probabilistic Model*	
Probability Model		*PMQDD*	*IPMQDD*
Similarity Distance	Name	vector similarity coefficient	vector inverse-equality similarity coefficient
	Symbol	κ^0	δ
	Definition*	$P_o(\mathbf{z} = 0)$	$\lvert 1 - 2P_o(\mathbf{z} = Z)\rvert$, $Z \in \{0,1\}$
Data and its inverse		Not Equivalent	Equivalent
Probability Model Equation		$1 - (\kappa^0)^p$	$1-(½)^p((1+\delta)^p + (1-\delta)^p)$
Expected number of mappings		$1 / (1 - \kappa^0)$	$[4 / (1 - \delta^2)] - 1$

* $\mathbf{z} = \mathbf{v_1} \oplus \mathbf{v_2}$, where $\mathbf{v_1}$ and $\mathbf{v_2}$ are the two vectors being matched.

5 Discussion

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 100 units x 100 units x 100 units. Each cubic unit can be considered as a voxel. If the object fills a given voxel then it is assigned a value of 1, otherwise the value of the voxel is 0. The entire cube containing the object and empty space is referred to as a model. Hence, each model is represented by $1x10^6$ 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,

- 6 of the 10 models are very similar; models 1 thru 6. The remaining 4 models, models 7 thru 10, are exact inverse instances of the first 4 models, respectively.
- 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 κ^0 for the set are in the range of 0 – 0.999, while values of δ 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_κ.
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_δ.

Table 2 shows the results of dissimilarity detection using the two probabilistic models. For each model pair, κ^0, $E(\kappa^0)$ and MDN_κ for *PMQDD* is displayed. This is followed by δ, $E(\delta)$ and MDN_δ 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 δ 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 found 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_κ for this model is 5.4%. The error between the predicted number of mappings $E(\delta)$ and actual number of mappings MDN_δ for this model is 1.5%.

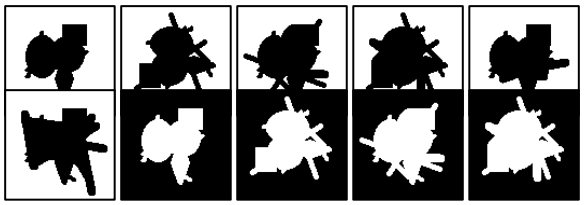

Fig. 5. The 10 models used for testing. From left to right and top to bottom: Models '1' to '10'.

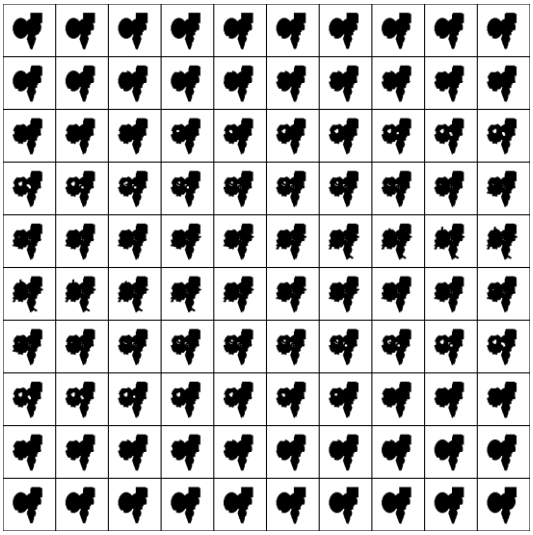

Fig. 6. The 100 slices of model 1.

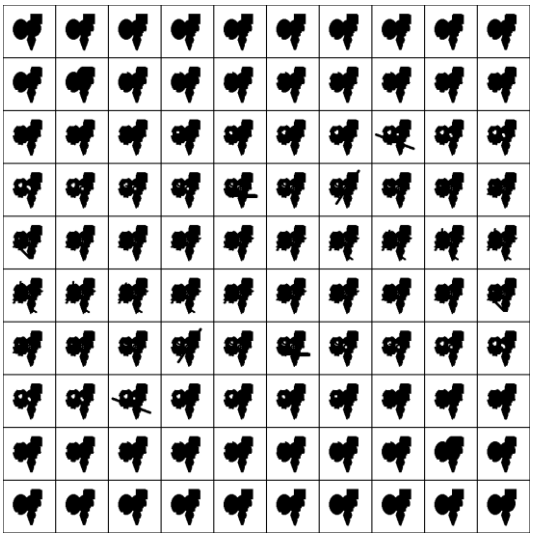

Fig. 7. The 100 slices of model 3.

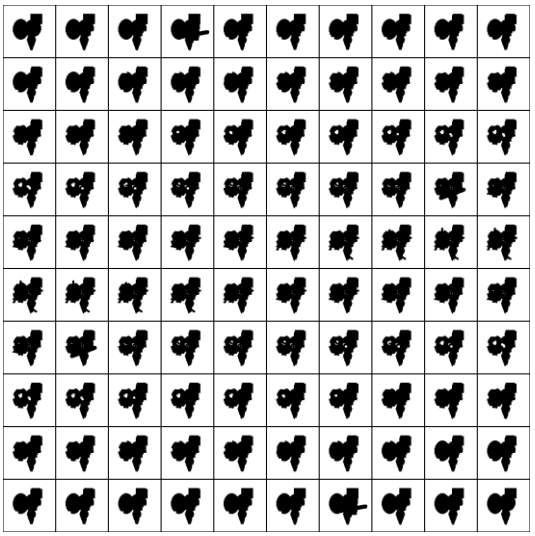

Fig. 8. The 100 slices of model 6.

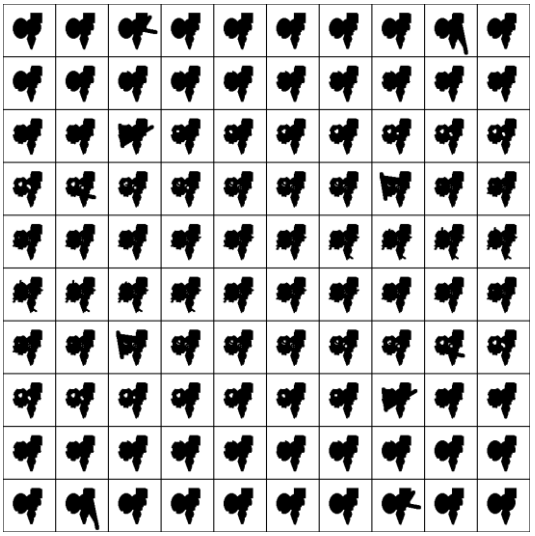

Fig. 9. The 100 slices of model 7.

Table 2: Dissimilarity detection results for the Model Set.

Model pairs		*PMQDD*			*IPMQDD*		
		κ^0	$E(\kappa^0)$	MDN_κ	δ	$E(\delta)$	MDN_δ
1	2	0.839	6.2	5.9	0.679	6.4	6.3
1	3	0.998	634	624	0.997	634	612
1	4	0.840	6.2	6.5	0.679	6.4	6.5
1	5	0.999	952	905	0.998	953	811
1	6	0.997	312	323	0.994	312	304
1	7	0.000	1.0	1.0	1.000	∞	*L*
1	8	0.161	1.2	1.2	0.679	6.4	6.1
1	9	0.002	1.0	1.0	0.997	634	622
1	10	0.160	1.2	1.2	0.679	6.4	6.4
2	3	0.840	6.2	6.2	0.679	6.4	6.5
2	4	0.999	1012	963	0.998	1012	887
2	5	0.840	6.2	6.3	0.680	6.4	6.7
2	6	0.838	6.2	6.4	0.677	6.4	6.3
2	7	0.161	1.2	1.2	0.679	6.4	6.6
2	8	0.000	1.0	1.0	1.000	∞	*L*
2	9	0.160	1.2	1.2	0.679	6.4	6.2
2	10	0.001	1.0	1.0	0.998	1012	912
3	4	0.840	6.2	6.0	0.680	6.4	6.2
3	5	0.997	381	363	0.995	381	378
3	6	0.995	209	206	0.990	209	199
3	7	0.002	1.0	1.0	0.997	634	649
3	8	0.160	1.2	1.2	0.679	6.4	6.6
3	9	0.000	1.0	1.0	1.000	∞	*L*
3	10	0.160	1.2	1.2	0.680	6.4	6.3
4	5	0.840	6.3	6.4	0.680	6.4	6.8
4	6	0.839	6.2	6.4	0.677	6.4	6.4
4	7	0.160	1.2	1.2	0.679	6.4	6.5
4	8	0.001	1.0	1.0	0.998	1012	939
4	9	0.160	1.2	1.2	0.680	6.4	6.3
4	10	0.000	1.0	1.0	1.000	∞	*L* ML
5	6	0.996	235	238	0.992	235	244
5	7	0.001	1.0	1.0	0.998	953	907
5	8	0.160	1.2	1.2	0.680	6.4	6.6
5	9	0.003	1.0	1.0	0.995	381	386
5	10	0.160	1.2	1.2	0.680	6.4	6.4
6	7	0.003	1.0	1.0	0.994	312	321
6	8	0.162	1.2	1.2	0.677	6.4	6.6
6	9	0.005	1.0	1.0	0.990	209	211
6	10	0.161	1.2	1.2	0.677	6.4	6.3
7	8	0.839	6.2	6.4	0.679	6.4	6.2
7	9	0.998	634	593	0.997	634	631
7	10	0.840	6.2	6.0	0.679	6.4	6.5
8	9	0.840	6.2	6.0	0.679	6.4	6.3
8	10	0.999	1012	974	0.998	1012	980
9	10	0.840	6.2	6.5	0.680	6.4	6.6

* A value of *L* for MDN_δ indicates that the number of mappings attempted to detect dissimilarity reached the limit without detecting dissimilarity.

Fig. 10. The 100 slices of model 8.

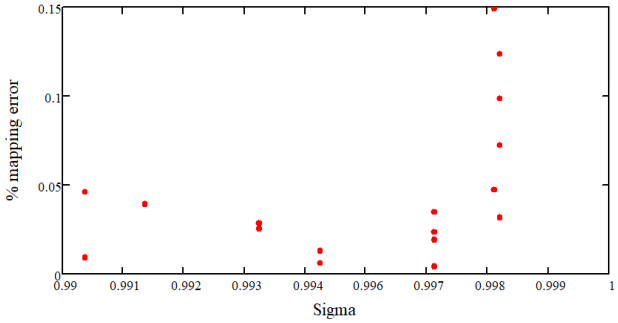

Fig. 11. A plot of % mapping error between E(δ) an MDN_δ vs. δ for model pairs with δ > 0.99..

An overall summary of results:

- The error between $E(\kappa^0)$ and MDN_κ 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_δ 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 δ between the models increases so does the error. This can be seen from Fig. 11 which shows a plot of MDN_δ vs. δ for model pairs with $\delta > 0.99$. As δ approaches 1 the error increases greatly. This is expected due to the sensitivity of mappings with high δ.

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* ($MDN_\kappa = 1$ 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* (κ^0), while *IPMQDD* is based on the *vector inverse-equality similarity coefficient* (δ). δ differs from κ^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:

[1] Mustafa, Adnan A., "Quick Matching of Big Binary Data: A Probabilistic Approach", *International Journal of Science and Technology*, Vol.9, No.28, 2016, pp. 1-11. DOI: 10.17485/ijst/2016/v9i28/97355.

[2] Brusco, M., Cradit, J.D. and Steinley, D., "A comparison of 71 binary similarity coefficients: The effect of base rates", *Plos one 16*, no. 4, 2021, pp. 1-19: e0247751.

[3] Consonni, V., Todeschini, R., "New similarity coefficients for binary data", *Match-Commun. Math. Comput. Chem.*, 68, (2), 2012, pp. 581–589.

[4] Lewis, D., Janeja, V., "An empirical evaluation of similarity coefficients for binary valued data", *IGI Global*, 2011, pp. 44–66.

[5] Choi, S., Cha, S., Tappert, C., "A survey of binary similarity and distance measures", *J. Systemics, Cybern. Inform.*, 8, (1), 2010, pp. 43–48.

[6] Jaccard, P., "Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines", *Bull. Soc. Vaudoise des Sci. Nat.*, 37, 1901, pp. 241–272.

[7] Sokal, R. and Michener, C., "A statistical method for evaluating systematic relationships", *Bull. Soc. Univ. Kansas*, 1958, 38, pp. 1409–1438.

[8] Hamming, R., "Error detecting and error correcting codes", *Bell Syst. Tech. J.*, 29, (2), 1950, pp. 147–160.

[9] Sidorov, G., Gelbukh, A., Gómez-Adorno, H., et al., "Soft similarity and soft cosine measure: similarity of features in vector space model", *Comput. Sist.*, 18, (3), 2014, pp. 491–504.

[10] Montgomery, D., Runger, G., *Applied statistics & probability for engineers*, John Wiley, 6th Edn., 2014.

[11] Cover, T., Thomas, J., *Elements of information theory*, John Wiley & Sons, New York, 2012.

[12] Tomaževič, D., Likar, B., Pernušj, F., "Multi-feature mutual information image registration", *Image Anal. Stereol.*, 31, 2012, pp. 43–53.

[13] Mustafa, Adnan A., "A Probabilistic Binary Similarity Distance for Quick Image Matching", *IET Journal on Image Processing*, 12 (10), 2018, pp. 1844-1856.

[14] Zomaya, Albert Y., and Sherif Sakr, eds. "Handbook of big data technologies". 2017, pp. 978-983.

[15] Huang Y., Zhu F., Yuan M., Deng K., Li Y., Ni B., Dai W., Yang Q., Zeng J., "Telco churn prediction with big data". In *Proceedings of the*

2015 ACM SIGMOD international conference on management of data, 2015, pp. 607-618.

[16] Esposito, C., Ficco, M., Palmieri, F., Castiglione, A., "A knowledge-based platform for big data analytics based on publish/subscribe services and stream processing", *Knowledge-Based Systems*, 79, 2015, pp. 3-17.

[17] Xu, J., Deng, D., Demiryurek, U., Shahabi, C., Schaar, M., "Mining the Situation: Spatiotemporal Traffic Prediction with Big Data", *IEEE Journal on Selected Topics in Signal Processing*, 9 (4), 7001625, 2015, pp. 702-715.

[18] Artikis, A., Etzion, O., Feldman, Z., Fournier, F., "A Tutorial: Event processing under uncertainty", in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*, 2012, pp. 32-43.

[19] Christophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., Stefanidis, K., "An Overview of End-to-End Entity Resolution for Big Data", *ACM Computing Surveys*, 53, (6), 2021, pp. 1-42.

[20] Mehmood, R., Meriton, R., Graham, G., Hennelly, P., "Exploring the influence of big data on city transport operations: a Markovian approach", *International Journal of Operations and Production Management*, 37 (1), pp. 75-104.

[21] Wang, X., Yang, L.T., Kuang, L., Zhang, Q., Deen, M.J., "A Tensor-Based Big-Data-Driven Routing Recommendation Approach for Heterogeneous Networks", *IEEE Network*, 33(1), 8610430, 2019, pp. 64-69.

[22] Wang, K., Shao, Y., Shu, L., Zhu, C., Zhang, Y., "Mobile big data fault-tolerant processing for ehealth networks", *IEEE Network*, 30(1), 7389829, 2016, pp. 36-42.

[23] Vogelstein, J., Conroy, J., Lyzinski, V., Vogelstein, R., Priebe, C., "Fast Approximate Quadratic programming for graph matching", *PLoS ONE*, 10 (4), e0121002, 2015.

[24] Guo, L., Ning, Z., Hou, W., Hu, B., Guo, P., Quick Answer for Big Data in Sharing Economy: Innovative Computer Architecture Design Facilitating Optimal Service-Demand Matching, *IEEE Transactions on Automation Science and Engineering*, 15 (4), 8372939, 2018, pp. 1494-1506.

[25] Fan, W., Geerts, F., Libkin, L., "On scale independence for querying big data", In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 2014, pp. 51-62.

[26] Yang, J., Jiang, B., Li, B., Tian, K., Lv, Z., "A Fast Image Retrieval Method Designed for Network Big Data", *IEEE Transactions on Industrial Informatics*, 13, (5), 7831461, 2017, pp. 2350-2359.

[27] Yang, K., Liu, R., Sun, Y., Yang, J., Chen, X., "Deep Network Analyzer (DNA): A Big Data Analytics Platform for Cellular Networks", *IEEE Internet of Things Journal*, 4 (6), 7733158, 2017, pp. 2019-2027.

[28] Zhang, K., Chen, K., Fan, B., "Massive picture retrieval system based on big data image mining", *Future Generation Computer Systems*, 121, 2021, pp. 54-58.

[29] Havens, T., Bezdek, J., Leckie, C., Hall, L., Palaniswami, M., "Fuzzy c-Means algorithms for very large data", *IEEE Transactions on Fuzzy Systems*, 20, (6), 6205366, 2012, pp. 1130-1146.

[30] Mustafa, Adnan A., "A Probabilistic Model for Random Binary Image Mapping", *WSEAS Transactions on Systems and Control*, Vol. 12, 2017, Art. #34, pp. 317-331, Dec. 2017.

[31] Wang, J., Liu, W., Kumar, S., Chang, "Learning to hash for indexing big data- A survey", *Proceedings of the IEEE*, 104, (1), 7360966, 2015, pp. 34-57.

Appendix

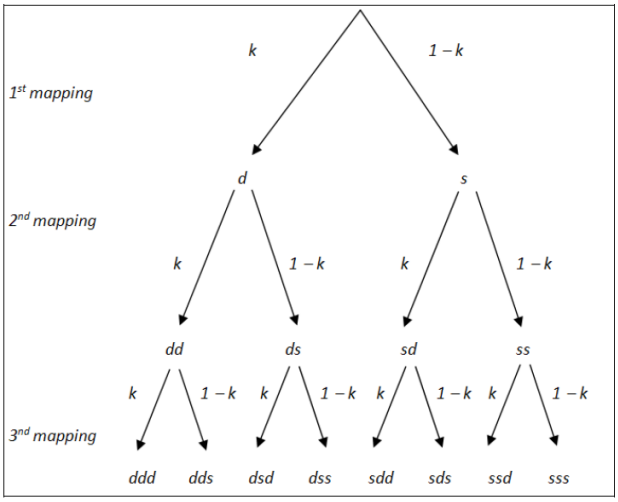

Fig. 2. Binary vector mappings.

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 $0 \le k \le 1$. 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, $k(1 - k)$, $k(1 - k)$ and $(1 - k)^2$, 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 [] given by,

$$\varphi(X = x, p, k) = \binom{p}{x} k^x (1-k)^{p-x}$$
$$x = 0, 1 \ldots p \qquad (12)$$

where X is a random variable denoting the number of times d occurs in p mappings and φ 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:

$$S = \{s, ss, sss, ssss, \ldots\} \qquad (13)$$

$$I = \{d, dd, ddd, dddd, \ldots\} \qquad (14)$$

$$M = \{sd, ssd, dss, dds, \ldots\} \qquad (15)$$

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,

$$\Pr(S,p,k) = \varphi(X{=}0,p,k) = (1-k)^p$$
$$p > 0 \qquad (16)$$

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

$$\Pr(D,p,k) = \varphi(0 \le X < p, p, k) = 1 - \varphi(X = p, p, k) \qquad (17)$$

Hence,

$$\Pr(D,p,k) = 1 - (1-k)^p$$
$$p = 1, 2, \ldots \text{ and } 0 \le k \le 1 \qquad (18)$$

But by its definition we see that,

$$(1-k) = \kappa^0(\mathbf{u},\mathbf{v}) = \varphi((\mathbf{z} = (\mathbf{u} \oplus \mathbf{v})) = 0) \qquad (19)$$

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

$$Pr_{QDD}(p, \kappa^0) = 1 - (\kappa^0)^p$$
$$p = 1, 2, \ldots \text{ and } 0 \le \kappa^0 \le 1 \qquad (7)$$

A.2 Proof of Pr_{IQDD} Equation

Since we are interested in the probability of occurrence of dissimilar vectors, then from (12),

$$\Pr(k, p) = \varphi(0 < X < p, p, k) = \sum_{x=1}^{p-1} \binom{p}{x} k^x (1-k)^{p-x} \qquad (20)$$

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

$$\Pr(k, p) = 1 - \left(\varphi(X = 0, p, k)) + \varphi(X = p, p, k)\right) \qquad (21)$$

which simplifies to,

$$\Pr(k, p) = 1 - \left((1-k)^p + k^p\right) \qquad (22)$$

But,

$$\kappa^0 = (\delta + 1) / 2 \qquad (23)$$

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

$$Pr_{IQDD}(p,\delta) = 1 - (½)^p \cdot ((1+\delta)^p + (1-\delta)^p) \qquad (10)$$