ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

Efficiency Analysis of Jaccard Similarity in Probabilistic Distribution Model

Jieru Zhang

Beijing Xinfeng Aerospace Equipment Co. Ltd, Intelligent Equipment and Technology Research Laboratory, Beijing, China
17801053406@163.com

Abstract: *The inner probabilistic properties of the big data have a great impact on the performance of pattern recognition systems. Jaccard similarity (JS) is a most popular statistic metric used for calculating the similarity of objects in feature extraction process. The paper combines JS with probabilistic distribution model to explore the effect of the inner properties of big data. It deduced the generalized form of JS for probabilistic model and determined the calculation method of JS for power-law and exponential distribution. Experiment observations showed that power-law distribution has higher JS than the correspondent exponential distribution, which denotes that power-law probabilistic structure is a more efficient probability structure. The original normalized data in MNIST database exhibited a more power-law-like distribution and the randomly translated data exhibited a more exponential-like distribution. The MNIST data with power-law-like property has higher JS and are more efficient comparing to the translated data. Thus, these observations provide possible guidelines for efficient information coding and processing methods.*

Keywords: *Jaccard similarity, Power-law distribution, Exponential distribution, Efficiency analysis*

1. Introduction

Deep learning and big data have become more and more attractive in both basic sciences and practical applications. Most of the studies target to improve the training and validation accuracy of machine learning to actual problems such as image recognition[1–3] and natural language processing (NLP)[4–6]. There are also many types of research which target to understand the intrinsic mechanisms of these machine learning systems which are able to solve very complex problems but are difficult to understand[7–9]. In addition, similarity search based on big data is also an important problem in many multimedia applications[10–12]. Furthermore, extracting essential features of original data is also a hot technique in data mining area[13,14]. However, though the big data is the foundation of these pattern recognition studies, the inner probabilistic properties of the data are lack of attention. For example, which kind of probabilistic properties do the original data have will have a higher accuracy in machine learning process? Which kind of probability structure that the extracted feature have can achieve classification better?

Jaccard similarity (JS) is a useful metric in evaluating the similarity of two sets and is widely used in many areas, such as NLP[15] and locality-sensitive hashing (LSH)[11]. Though JS has a strong mathematical basis, it's indirect for researchers to evaluate a whole system which consists of many objects[16–18]. A possible approach is to calculate the JS of all possible object pairs in the system. The limitation of this approach lies in the abandon of statistical and probabilistic feature in the original data. The most widely used statistical and probabilistic method in machine learning is Bayesian learning technique which is based on Bayes' theorem[19–22]. Nevertheless, most of the Bayes-related studies focus on updating the probability about inference and neglect the probability structure of the original data by assuming the distribution in advance[19,20]. In literature and applications, exponential distribution is a very popular function when modeling probabilistic systems[23–25]. However, exponential distribution cannot properly model inhomogeneous systems in which there are usually several hot spots which are connected to most of the other nodes in system and these few hubs dominate the function of the whole system. Inhomogeneous systems are widespread phenomenon and can be perfectly modeled by power-law distribution[26–30]. Power-law distribution is similar to exponential distribution in linear-linear systems and it's necessary to distinguish them when modeling probabilistic systems.

We combine JS with probabilistic distribution model in this study with the goal to explore the effect

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

of the inner properties of big data on pattern recognition systems.

The rest of this study is organized as follows: Section 2 introduces the definition of JS and its generalization to probabilistic distribution model. Section 3 cross-validates the generalization of JS by analytical and numerical approach. Section 4 describes a practical application to MNIST database. Conclusions are presented in Section 5.

2. Jaccard similarity

2.1. Classical Jaccard similarity

Jaccard similarity (JS) is a most popular statistic metric used for calculating the similarity and dissimilarity of sample sets. The JS of two sets A and B is defined as

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

In practical applications, JS is a useful indices for objects with n binary attributes. For two objects A and B, and denote the count of attributes when both two objects have the value of 1 and 0, respectively. The attributes when both two objects have the value of 1 and 0 are called as positive matching attributes and negative matching attributes, respectively. Denotes the number of attributes when A is 1 and B is 0. Denotes the number of attributes when A is 0 and B is 1. According to the properties of set theory, we have

$$|A \cap B| = M_{11}$$

$$|A \cup B| = M_{11} + M_{10} + M_{01}$$

Then

$$J(A,B) = \frac{M_{11}}{M_{11} + M_{10} + M_{01}}$$

A most critical property of JS is that it does not take into account of negative matching attributes comparing to the simple matching coefficient method. For some applications, positive and negative matching attributes have asymmetric information. In the Market Basket Analysis example, two customers may have bought several same products in a supermarket, but there are also more products they do not buy. In brain neuroscience example, there are several neurons which respond to two external stimuli, but there are more neurons which keep silence to both stimuli. In NLP systems, two sentences may consist of several same words, but there are more words which do not occur in both sentences. In these cases, the negative matching attributes have no meaningful contribution to the measurement of similarity or diversity.

In literature, there is also a kind of synonyms for JS in which the similarity is given over Boolean algebra operations. $A = (A_1, A_2, \cdots, A_n)$, $B = (B_1, B_2, \cdots, B_n)$, where $A_k = \{0,1\}$, $B_k = \{0,1\}$.

The JS of A and B is defined as

$$J(A,B) = \frac{\sum_{k=1}^{n} A_k \wedge B_k}{\sum_{k=1}^{n} A_k \vee B_k}$$

2.2. Generalized Jaccard similarity

In mathematical logic, the values of the variables in Boolean algebra are the truth values true and false, usually denoted as 1 and 0 respectively. Though Boolean algebra is the foundation of digital electronics and computers, as two-valued logic, it is difficult to describe complex actual systems in applications by Boolean algebra. When describing practical systems, there are usually multiple possible values for an attribute. For example, a word may occur several times in a sentence or a paragraph in the NLP systems and multi-valued logic would be more proper to this scenario.

As a result, the calculation of JS also needs to be extended to adapt to actual applications basing on multi-valued logic. Usually, there is a naturally existing or user-defined threshold for the value range of each attribute. Such as the value of each pixel in an 8-bit image would not exceed 256 or the occurrence times of a sentence would not exceed the total words of this sentence. The threshold vector of n attributes is denoted as

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

$$T = (t_1, t_2, \cdots, t_n)$$

Two objects A and B are denoted as

$$A = (A_1, A_2, \cdots, A_n)$$

$$B = (B_1, B_2, \cdots, B_n)$$

Where $0 \leq A_k, B_k \leq t_k$. The objects A and B after normalization by threshold vector T are denoted as a and b, respectively. And it would be seen as the occurrence probability or intensity ration of the *k*-th attribute.

$$a = (a_1, a_2, \cdots, a_n)$$

$$b = (b_1, b_2, \cdots, b_n)$$

$$a_k = \frac{A_k}{t_k}$$

$$b_k = \frac{B_k}{t_k}$$

$$J(A, B) = J(a, b) = \frac{\sum_{k=1}^{n} a_k \wedge b_k}{\sum_{k=1}^{n} a_k \vee b_k}$$

Where $0 \leq a_k, b_k \leq 1$. According to probability theory, we have

$$a_k \wedge b_k = a_k \cdot b_k$$

$$a_k \vee b_k = 1 - (1 - a_k) \cdot (1 - b_k) = a_k + b_k - a_k \cdot b_k$$

$$J(A, B) = \frac{\sum_{k=1}^{n} a_k \cdot b_k}{\sum_{k=1}^{n} a_k + b_k - a_k \cdot b_k}$$

2.3. Jaccard similarity of probabilistic vector

There are many applications in which JS is used to calculate the similarity of two objects such as semantic understanding and locality sensitive hashing. In addition, the probabilistic basis and the tables of significant values of JS are also proposed. However, there are few works which explore JS of probabilistic models. For example, uncertainty is a primary property of brain neural circuit and one external stimulus would evoke several different neural responses. As a result, the neural responses to an external stimulus is a probabilistic model[31]. We can calculate the JS of stimulus-evoked neural responses, but how can the relationship between stimulus and the response of neural circuit be evaluated?

For a probability vector

$$Prob = (p_1, p_2, \cdots, p_n)$$

Two object vectors and are generated according to the probability vector *Prob*, where $\mathrm{P}(x_k = 1) = p_k$, $\mathrm{P}(y_k = 1) = p_k$. These settings can be seen as the k-th neuron would respond to the given stimulus with the probability of in a monitored brain neural circuit, or the k-th word would have the occurrence intensity of in a given class of natural context.

$$J(x, y) = \frac{\sum_{k=1}^{n} p_k \cdot p_k}{\sum_{k=1}^{n} p_k + p_k - p_k \cdot p_k}$$

$$J(x, y) = \frac{\sum_{k=1}^{n} p_k^2}{\sum_{k=1}^{n} 2p_k - p_k^2}$$

Actually, objects x and y are both generated according to probability vector *Prob*. is a function of *Prob*. As a result, we can define the JS of probabilistic vector *Prob* as

$$J(Prob) = \frac{\sum_{k=1}^{n} p_k^2}{\sum_{k=1}^{n} 2p_k - p_k^2}$$

JS of probabilistic vector would provide insights into information encoding process, such as in neural encoding process and locality sensitive hashing function. Probability vector *Prob* can be seen as the statistical property of many codes which correspond to the same tag, such as the same external stimulus

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

or the same type of context. We expect more similar object vectors because these codes have the same tag. As a result, a higher value of denotes a more efficient information processing mechanism which can preserve the similarity information better.

2.4. Jaccard similarity of probabilistic distribution model

Assume the probability vector follows a distribution of $f(p)$. We have

$$\sum_{k=1}^{n} p_k = \int_{\varepsilon}^{1} p\, f(p) dp$$

$$\sum_{k=1}^{n} p_k^2 = \int_{\varepsilon}^{1} p^2 f(p) dp$$

Where $\varepsilon \to 0$. Then

$$J(Prob) = \frac{\int_{\varepsilon}^{1} p^2 f(p) dp}{2\int_{\varepsilon}^{1} p\, f(p) dp - \int_{\varepsilon}^{1} p^2 f(p) dp}$$

Here we have generalized the calculation of JS to the continuous probabilistic distribution model. In literature and many applications, Power-law and exponential distributions are two most widely used probabilistic distributions which are used to model the practical systems. In the following context, we focus on the analysis of JS of Power-law and exponential distributions.

2.4.1. Power-law distribution

Power-law distribution is also known as Pareto distribution. Pareto principle denotes that 80% of social wealth is held by 20% of the population, calling '80-20 rule'[32]. Though power-law distribution is originally observed in economics, it's a general rule in many areas, such as the world-wide-web, the social relationship, the collaboration between scientists, the airline networks, and the brain neural networks. A key property of power-law network lies in high random error tolerance[26,33]. It's reported that brain neural network exhibits power-law properties when the subject implements tasks. In addition, the cultured neural networks in vitro show a similar phenomenon. In other words, neural circuit tends to encode information into the codes which have power-law property. Neural circuits are natural locality sensitive hashing function which assigns similar response to similar stimulus when animal perceive the environment. Here we check the JS properties of the power-law distribution.

Consider the scenario that the probability vector follows the Power-law distribution function

$$f(x) = \mathrm{c} \cdot x^{-r}$$

Where $\mathrm{x} \in (\varepsilon, 1]$. Then

$$J(Pow) = \frac{\int_{\varepsilon}^{1} \mathrm{x}^2 \cdot \mathrm{c} \cdot \mathrm{x}^{-\mathrm{r}} \mathrm{dx}}{2\int_{\varepsilon}^{1} \mathrm{x} \cdot \mathrm{c} \cdot \mathrm{x}^{-\mathrm{r}}\, \mathrm{dx} - \int_{\varepsilon}^{1} \mathrm{x}^2 \cdot \mathrm{c} \cdot \mathrm{x}^{-\mathrm{r}} \mathrm{dx}}$$

$$J(Pow) = \frac{\int_{\varepsilon}^{1} \mathrm{x}^2 \cdot \mathrm{x}^{-\mathrm{r}}\, \mathrm{dx}}{2\int_{\varepsilon}^{1} \mathrm{x} \cdot \mathrm{x}^{-\mathrm{r}}\, \mathrm{dx} - \int_{\varepsilon}^{1} \mathrm{x}^2 \cdot \mathrm{x}^{-\mathrm{r}} \mathrm{dx}}$$

$$J(Pow) = \frac{1}{\frac{2\int_{\varepsilon}^{1} \mathrm{x} \cdot \mathrm{x}^{-\mathrm{r}} \mathrm{dx}}{\int_{\varepsilon}^{1} \mathrm{x}^2 \cdot \mathrm{x}^{-\mathrm{r}}\, \mathrm{dx}} - 1}$$

Power-law distribution is a straight line in double logarithmic (log-log) coordinate systems and the decay slope in log-log systems is denoted as $logD(Pow)$. For two different variables $x_1, x_2 \in (\varepsilon, 1]$, $x_1 \neq x_2$,

$$logD(Pow) = -\frac{\log\big(f(x)\big) - \log\big(f(x)\big)}{\log(x_2) - \log(x_1)}$$

$$logD(Pow) = -\frac{\log(\mathrm{c} \cdot {x_2}^{-r}) - \log(\mathrm{c} \cdot {x_1}^{-r})}{\log(x_2) - \log(x_1)}$$

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

$$logD(Pow) = r$$

2.4.2. Exponential distribution

Exponential distribution is a universal probabilistic model. Consider the scenario that the probability vector follows the exponential distribution function

$$g(x) = a \cdot e^{-b \cdot x}$$

Where $x \in (\varepsilon\ 1]$. Then

$$J(Exp) = \frac{\int_{\varepsilon}^{1} x^2 \cdot a \cdot e^{-b \cdot x}\, dx}{2\int_{\varepsilon}^{1} x \cdot a \cdot e^{-b \cdot x} dx - \int_{\varepsilon}^{1} x^2 \cdot a \cdot e^{-b \cdot x} dx}$$

$$J(Exp) = \frac{\int_{\varepsilon}^{1} x^2 \cdot e^{-b \cdot x} dx}{2\int_{\varepsilon}^{1} x \cdot e^{-b \cdot x} dx - \int_{\varepsilon}^{1} x^2 \cdot e^{-b \cdot x} dx}$$

$$J(Exp) = \frac{1}{\frac{2\int_{\varepsilon}^{1} x \cdot e^{-b \cdot x} dx}{\int_{\varepsilon}^{1} x^2 \cdot e^{-b \cdot x}\, dx} - 1}$$

Exponential distribution takes the form of curve line in log-log coordinate systems and it has a larger slope magnitude when the abscissa gets bigger. However, there is a relatively linear range in log-log systems when abscissa is small and the decay slope in log-log systems is denoted as $logD(Exp)$. This linear range was denoted as in linear-linear systems.

$$logD(Exp) = -\frac{\log\big(g(x_2)\big) - \log\big(g(x_1)\big)}{\log(x_2) - \log(x_1)}$$

$$logD(Exp) = -\frac{\log(a \cdot e^{-b \cdot x_2}) - \log\big(a \cdot e^{-b \cdot x_1}\big)}{\log(x_2) - \log(x_1)}$$

$$logD(Exp) = \frac{b(x_2 - x_1) \cdot \log e}{\log\left({}^{x_2}/_{x_1}\right)}$$

2.5. Comparison of Jaccard similarity

Both Power-law and exponential distributions decay very fast when the x-coordinate is small and it slows down when the x-coordinate gets bigger in linear-linear systems. Most of the features of this two distributions in linear-linear systems can be characterized by the decay slope in log-log systems. Before comparing the JS of this two distributions, we set

$$logD(Pow) = logD(Exp)$$

$$\frac{b(x_2 - x_1) \cdot \log e}{\log\left({}^{x_2}/_{x_1}\right)} = r$$

$$b = \frac{r \cdot \log\left({}^{x_2}/_{x_1}\right)}{(x_2 - x_1) \cdot \log e}$$

In addition, we set the integrations of Power-law and exponential distributions in range to be equal which denote a foundation that the total number or intensity of positive attributes are equal in two distributions.

$$\int_{\varepsilon}^{1} x \cdot f(x)\, dx = \int_{\varepsilon}^{1} x \cdot g(x)\, dx$$

$$\int_{\varepsilon}^{1} x \cdot c \cdot x^{-r}\, dx = \int_{\varepsilon}^{1} x \cdot a \cdot e^{-b \cdot x}\, dx$$

Then the parameter b in exponential distribution is a function of r in Power-law distribution. As a result, both JS of Power-law and exponential distribution are the functions of parameter r. In addition, once the value of parameter c is determined, the value of a can also be calculated. Till now, we have

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

bring the JS of Power-law and exponential distribution into a comparable system.

3. Validation analysis of Jaccard similarity in probabilistic distribution model

3.1. Experiment settings

Firstly, we chose the lower limit when calculating the JS of Power-law and exponential distribution. We set which indicates for the consideration that 0.05 is a significant value in the significant test. In actual applications, this setting is reasonable and necessary. For example, after translating a large context into a vector of word occurrences, there would be many words which occur very few times and these words are likely to provide a very limited contribution to context classification in NLP systems. In another brain neuroscience example, there are many neurons which respond to multiply repeated stimuli with a very low probability. These cells are very likely to be uncorrelated to the given stimulus and their activity may be spontaneous or other stimulus correlated. As a result, setting a lower limit would have several advantages in practical analysis: 1) reducing the noise or target-uncorrelated factors, 2) decreasing the number of attributes and decreasing the computational complexity. Here we set $x_1 = \varepsilon = 0.05$, $x_2 = 0.2$. Decay slope in log-log systems with a step size of 0.1 in efficiency analysis of JS. All experiments were conducted in Matlab 2018a software.

3.2. Efficiency analysis of Jaccard similarity

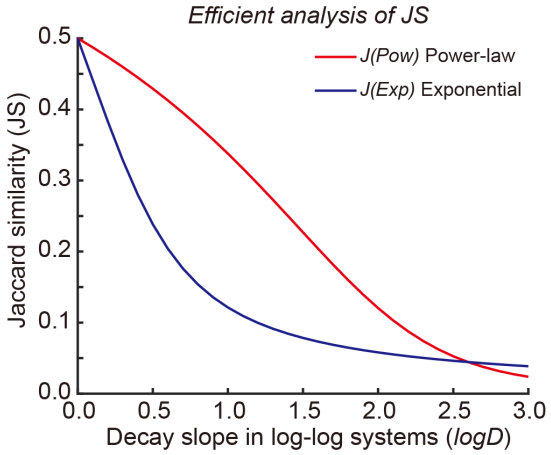

Figure 1: Efficiency analysis of Jaccard similarity. (Red and blue line denote JS of Power-law and exponential distribution, respectively.)

When $logD(Pow) = logD(Exp) = 0$, both Power-law and exponential distributions degenerate to a horizontal line which indicate that a system have equal number of attributes across all positive probability or intensity in the range of $(\varepsilon\ 1]$. The JS of Power-law and exponential distributions is 0.5 ($J(Pow) = J(Exp) = 0.5$) under this circumstance. In addition, this horizontal line distribution should be distinguished with the degenerated distribution of a point which indicates that all attributes have the same positive probability or intensity. The JS of both distributions is a decrease function of $logD$. However, the JS of Power-law distribution decrease more slowly than the JS of exponential distribution (Figure 1).

We calculated the ratio of JS for both two distributions in the same value of $logD$.

$$Ratio = \frac{J(Pow)}{J(Exp)}$$

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

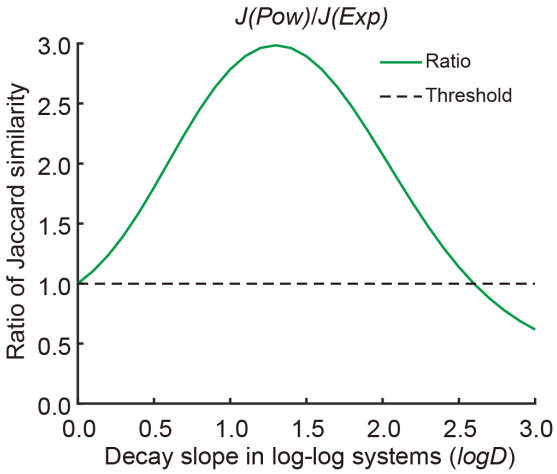

Figure 2: Jaccard similarity ratio. (The ratio is calculated as $J(Pow)/J(Exp)$. Green solid line denotes the ratio of JS. Black dotted line denotes the threshold of 1.0 which means $J(Pow) = J(Exp)$.)

The value of exhibited as a bell-shaped curve. When $logD = 2.60$, it is bigger than that when $logD \in (0\ 2.60)$. Furthermore, it reaches the maximum of 2.99 (Figure 2). In addition, two example distributions with and are illustrated in Figure 3. JS is the function of decay parameter *r* and *b*, and is uncorrelated to the linear parameter *c* and *a* for Power-law and exponential distributions, respectively. We chose the linear parameter value to ensure which will make it more convenient to compare the properties of two probabilistic distribution models. (Figure 2).

3.3. Numerical validation of efficiency analysis

Here we validated the phenomenon about JS of probabilistic distribution model by a numerical method that JS of Power-law distribution is bigger than exponential distribution when $logD \in (0\ 2.60)$. (Figure 3) We generated probability vectors with 10,000 attributes which follows Power-law and exponential distributions. It was set as 1.50 and 2.00 (Figure 4). Then object vectors *z* with 10,000 attributes were generated according to the determined probability vectors. According to the definition of classical JS, the value of these attributes are chosen from {0,1}.

$$Prob = (p_1, p_2, \cdots, p_{10000})$$

$$z = (z_1, z_2, \cdots, z_{10000})$$

Where $z_k \in \{0,1\}$.

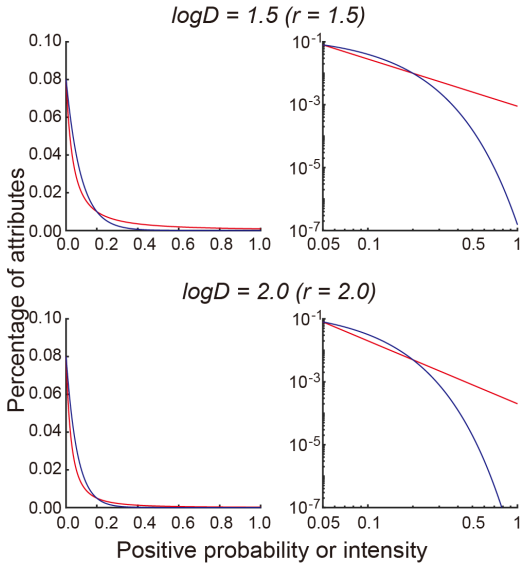

Figure 3: Example distributions in linear-linear and log-log systems. (Upper panels, $logD = 1.50$. Bottom panels, $logD = 2.00$.)

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

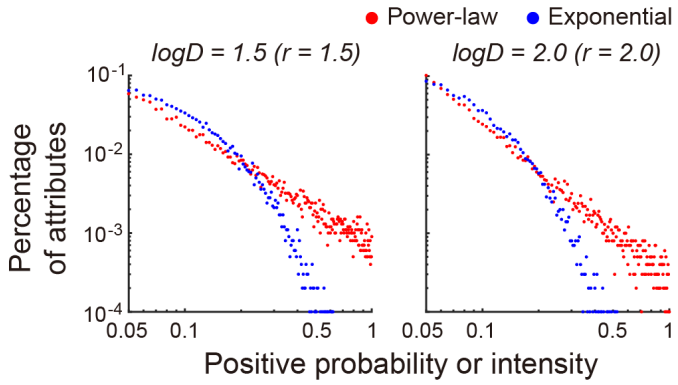

Figure 4: Distribution of generated probability vectors according to Power-law and exponential distribution.

When determining the value of z_k, we generated a random parameter which are evenly chosen from [0 1].

$$z_k = \begin{cases} 1 \text{ if } q_k \leq p_k \\ 0 \text{ else} \end{cases}$$

1,000 object vectors are generated independently for both two distributions. We calculated the pairwise JS of all 1,000 object vectors. When $logD = 1.50$, The numerical results of JS for Power-law distribution and exponential distribution are $J(Pow)$: and $J(Exp)$: 0.090 ± 0.006, respectively. Ratio of JS: 2.975 ± 0.213. When $logD = 2.00$, The numerical results of JS for Power-law distribution and exponential distribution are $J(Pow)$: and $J(Exp)$: 0.070 ± 0.006, respectively. Ratio of JS: 2.046 ± 0.198. Cross validation results show there are no significant difference between analytical and numerical results about the ratio of JS (Figure 5).

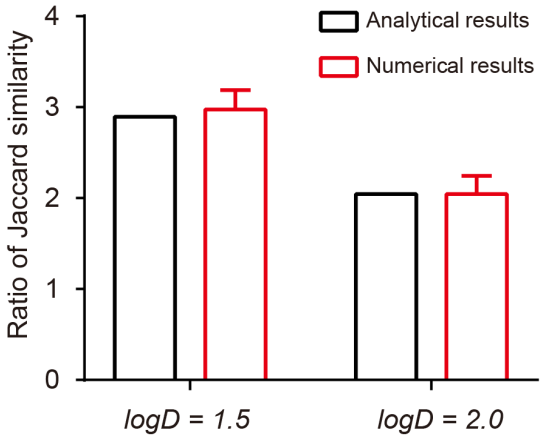

Figure 5: Cross validation of analytical and numerical results about the ratio of JS.

4. Practical application to MNIST database

The MNIST database of handwritten digits is a widely used benchmark for machine learning and pattern recognition techniques[34]. All the digits are size-normalized and centered in a 28×28 image. There are many studies which investigate classification methods and locality sensitive hashing methods basing MNIST database. However, the inner data structure and the global probabilistic distribution model underling the database was still unclear. For example, what distribution can properly model the size-normalized and centered images of digits and how random translation operations have effects on the similarity of image pairs?

Firstly, we implemented random translations to all 60,000 training examples. The translation range of both dimension was and the maximum translation step was 25% (7/28) of the image size (Figure 6). Secondly, we analyzed the probabilistic distribution models of both original and translated images according to labels. We changed the gray images into binary images by setting the threshold of 0 and got an averaged image by calculating the mean value of pixels over the binary images which have the

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

same label. We got 10 averaged 28×28 images which denote the positive probability of each pixel corresponding to 10 labels. All the pixels can be seen as 784 (28×28) attributes. Then the distribution of all these positive probability was analyzed.

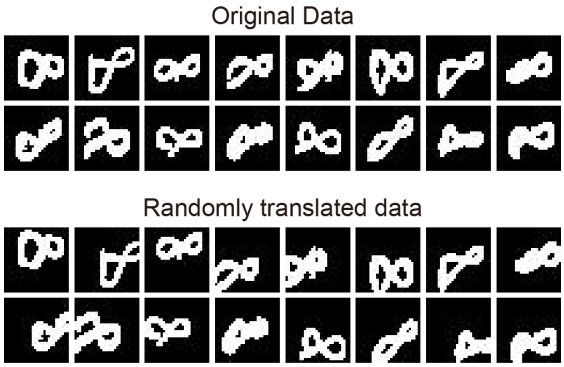

Figure 6: Examples of original and randomly translated handwritten digits data in MNIST database.

The percentage of attributes for both original and translated images decreases with the increase of positive probability. However, the original data exhibited a more power-law-like distribution and the randomly translated data exhibited a more exponential-like distribution in log-log systems. The distribution of original data in log-log systems can be well fitted by a line which is actually a Power-law distribution function with the decay slope of 0.608 in the range of [0.01 0.5] (R^2 = 0.925) (Figure 7).

We then implemented JS experiments over the original and translated images using nearest neighbor search procedure. We randomly chose 600 query images from the total 60,000 images (1%). For each query image, we find the top 1,200 (2%) nearest neighbors from the total images according to the value of JS. The distribution of JS of nearest neighbor for both original and randomly translated data can be well modeled by Gaussian distribution function: original $\mu = 0.472$, $\sigma^2 = 0.008$, $R^2 = 0.985$; translated $\mu = 0.329$, $\sigma^2 = 0.003$, $R^2 = 0.992$. Experimental results shown that original data which has power-law-like properties has significantly higher JS of nearest neighbor than the randomly translated data which exhibits exponential-like properties (Figure 8).

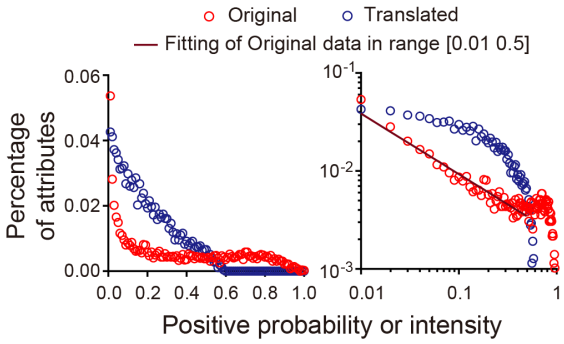

Figure 7: Probabilistic properties of original and randomly translated data.

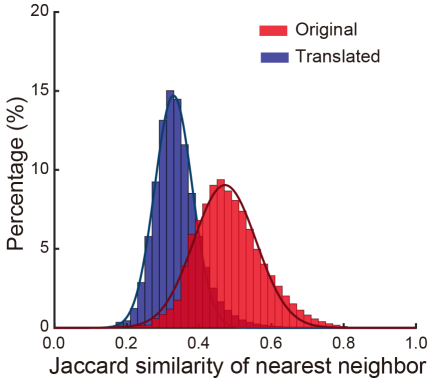

Figure 8: Distribution of Jaccard similarity of nearest neighbor for both original and randomly translated data.

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

5. Conclusion

Overall, this paper analyzed the efficiency of JS for probabilistic models in several ways. (1) The author deduced the generalization form of JS for probabilistic model and determined the calculation method of JS for Power-law and exponential distribution. (2) The author found the data with power-law probabilistic structure has higher JS comparing to a correspondent exponential distribution. We cross-validated this observation by analytical and numerical approach. (3) The author applied the generalized JS to investigate the probabilistic properties of MNIST database. The author found the original normalized data in MNIST exhibited a more power-law-like distribution and the randomly translated data exhibited a more exponential-like distribution. The data which have power-law-like properties have higher JS and are more efficient than the randomly translated data which exhibit exponential-like properties. This study provides possible guidelines for efficient information coding and processing methods.

References

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions, IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1–9, 2015.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein, ImageNet Large Scale Visual Recognition Challenge, Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, in International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[4] K. S. Tai, R. Socher, and C. D. Manning, Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks, Comput. Sci., vol. 5, no. 1, p. : 36., 2015.

[5] Y. Kim, Convolutional Neural Networks for Sentence Classification, Eprint Arxiv, 2014.

[6] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, A Convolutional Neural Network for Modelling Sentences, Eprint Arxiv, vol. 1, 2014.

[7] R. S. A. M. David Mascharka Philip Tran, Transparency by Design: Closing the Gap Between Performance and Interpretability in Visual Reasoning, IEEE Conf. Comput. Vis. Pattern Recognit., 2018.

[8] M. B. Ari S. Morcos, David G.T. Barrett, Neil C. Rabinowitz, On the importance of single directions for generalization, Int. Conf. Learn. Represent., 2018.

[9] S. Ritter, D. G. T. Barrett, A. Santoro, and M. M. Botvinick, Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study, in Proceedings of the 34 th International Conference on Machine Learning, 2017.

[10] S. Dasgupta, C. F. Stevens, and S. Navlakha, A neural algorithm for a fundamental computing problem., Science (80-.)., vol. 358, no. 6364, pp. 793–796, 2017.

[11] J. Ji, J. Li, S. Yan, Q. Tian, and B. Zhang, Min-Max Hash for Jaccard Similarity, in IEEE International Conference on Data Mining, 2014, pp. 301–309.

[12] A. Gionis, P. Indyk, and R. Motwani, Similarity Search in High Dimensions via Hashing, in International Conference on Very Large Data Bases, 1999, pp. 518–529.

[13] J. Yang, A. F. Frangi, J. Y. Yang, D. Zhang, and Z. Jin, KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition, IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 2, pp. 230–244, 2005.

[14] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, Feature extraction from faces using deformable templates, Int. J. Comput. Vis., vol. 8, no. 2, pp. 99–111, 1992.

[15] J. Singthongchai and S. Niwattanakul, A Method for Measuring Keywords Similarity by Applying Jaccard's, N-Gram and Vector Space, Lect. Notes Inf. Theory, vol. 1, no. 4, pp. 159–164, 2013.

[16] R. Real, Tables of significant values of Jaccard's index of similarity, Vet. Rec., vol. 22, no. 14, pp. 456–457, 1999.

[17] R. Real and J. M. Vargas, The Probabilistic Basis of Jaccard's Index of Similarity, Syst. Biol., vol. 45, no. 3, pp. 380–385, 1996.

[18] M. Levandowsky and D. Winter, Distance between Sets, Nature, vol. 239, no. 5368, p. 174, 1971.

[19] E. Mossel, N. Olsman, and O. Tamuz, Efficient Bayesian Learning in Social Networks with Gaussian Estimators, in Communication, Control, and Computing, 2017, pp. 425–432.

[20] C. K. Wen, S. Jin, K. K. Wong, J. C. Chen, and P. Ting, Channel Estimation for Massive MIMO Using Gaussian-Mixture Bayesian Learning, IEEE Trans. Wirel. Commun., vol. 14, no. 3, pp. 1356–1368, 2015.

ISSN 2616-5775 Vol. 6, Issue 2: 53-63, DOI: 10.25236/AJCIS.2023.060208

[21] H. Liu, J. Bo, H. Liu, and Z. Bao, Superresolution ISAR Imaging Based on Sparse Bayesian Learning, IEEE Trans. Geosci. Remote Sens., vol. 52, no. 8, pp. 5005–5013, 2014.
[22] M. E. Tipping, Sparse Bayesian Learning and Relevance Vector Machine, J. Mach. Learn. Res., vol. 1, no. 3, pp. 211–244, 2001.
[23] D. Kundu and R. D. Gupta, Generalized exponential distribution: Bayesian estimations, Comput. Stat. Data Anal., vol. 52, no. 4, pp. 1873–1883, 2008.
[24] W. J. Ma, J. M. Beck, P. E. Latham, and A. Pouget, Bayesian inference with probabilistic population codes, Nat. Neurosci., vol. 9, no. 11, pp. 1432–1438, 2006.
[25] R. D. Gupta and D. Kundu, Generalized exponential distribution: different method of estimations, J. Stat. Comput. Simul., vol. 69, no. 4, pp. 315–337, 2001.
[26] G. Zheng and Q. Liu, Scale-free topology evolution for wireless sensor networks, Comput. Electr. Eng., vol. 39, no. 6, pp. 1779–1788, 2013.
[27] A. Clauset, C. R. Shalizi, and M. E. J. Newman, Power-Law Distributions in Empirical Data, Siam Rev., vol. 51, no. 4, pp. 661–703, 2012.
[28] M. L. Goldstein, S. A. Morris, and G. G. Yen, Problems with fitting to the power-law distribution, Eur. Phys. J. B - Condens. Matter Complex Syst., vol. 41, no. 2, pp. 255–258, 2004.
[29] X. Gabaix, P. Gopikrishnan, V. Plerou, and H. E. Stanley, A theory of power-law distributions in financial market fluctuations, Nature, vol. 423, no. 6937, pp. 267–270, 2003.
[30] M. Levy and S. Solomon, New evidence for the power-law distribution of wealth. Physica A: Statistical Mechanics and its Applications, vol. 242, no. 1–2, pp. 90–94, 1997.
[31] Y. Tian, C. Yang, Y. Cui, et al., An excitatory neural assembly encodes short-term memory in the prefrontal cortex. Cell Rep., vol. 22, no. 7, pp. 1734–1744, 2018.
[32] M. E. J. Newman, Power laws, Pareto distributions and Zipf's law, Contemp. Phys., vol. 46, no. 5, pp. 323–351, 2005.
[33] Feng, Peijiang, Yuan, Yangzhen, Wang, Chen, and Zhang, The superior fault tolerance of artificial neural network training with a fault/noise injection- based genetic algorithm. Protein Cell, vol. 7, no. 10, pp. 735–748, 2016.
[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition. Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.