NI-DBSCAN: DBSCAN under Non-IID

Yikun Lv, He Jiang and Pinchen Pan

School of Computer Science & Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China

*Corresponding author e-mail: lyk20169@163.com

Abstract. DBSCAN (Density Based Spatial Clustering of Application with Noise) is an example of density-based clustering algorithm. Aiming at problem that DBSCAN algorithm assumes that the data are independent and identically distributed and the traditional distance formula is difficult to accurately calculate the similarity degree between categorical data. Density Based Spatial clustering algorithm of Application with Noise under Non-IID (NI-DBSCAN) is proposed. The unsupervised clustering problem of categorical data is dealt with by means of the Non-IID (non-independent and identical distribution) thought. Using coupling similarity to measure similarity can better reflect the "real relationship" between categorical data. The experimental results on the UCI dataset show that the algorithm can obtain satisfactory clustering results and improve the applicability and accuracy of the algorithm.

Keywords: Categorical Data Clustering, Coupling Similarity Matrix, DBSCAN Algorithm, Principle of Statistics

1. Introduction

With the rise of data mining, data clustering, as a common technique for data analysis, has been used in many fields, such as machine learning [1, 2], data mining [3-5], pattern recognition [6] and image analysis [7]. DBSCAN algorithm is a density-based clustering algorithm, which can detect clusters of arbitrary shape and can effectively identify discrete points.

Sunita [8] according to the density distribution of the data set in each dimension dynamically set Eps parameters, but still needed to input Minpts parameters, without realizing the full automation of clustering; Khan et al. [9] proposed that AD-DBSCAN algorithm adaptive clustering algorithm needs to specify the number of clusters in advance, so it cannot automatically identify the number of cluster classes. Although DBSCAN algorithm has made a lot of improvements, most of them assume that the data are independent and identically distributed. However, this assumption ignores implicit relationships between data objects, including explicit or implicit coupling relationships between data objects, between data object attributes, and between attribute values, which results in bias in experimental results. To solve this problem, we propose a DBSCAN under Non-IID (NI-DBSCAN). The idea of non-independent and identical distribution learning was proposed in 2011[10, 11]. Subsequently, this idea has been applied to different fields, such as coupling relationship [12-14], anomaly detection [15, 16], recommendation system [17, 18], and data mining [19-21] and so on.
In this paper, we propose a Density Based Spatial Clustering of Application with Noise under Non-IID (NI-DBSCAN). The coupling similarity of categorical data under the condition of non-independent and distribution is used to calculate the similarity between data object and attribute value, and output it in the form of a matrix. Scatter diagram is used to intuitively represent the distribution of data and observe it to find the boundary point of coupling similarity between clusters and data objects among clusters, which is taken as the neighborhood interval lower limit value $Eps_1$. At the same time, find the right k-nearest (k takes the value of Minpts) value of the neighborhood interval lower limit value $Eps_1$ and use the maximum value as the neighborhood interval upper limit value $Eps_2$. Select the parameter Minpts value according to the actual situation. The clustering process fully considers the real relationship between the data, and the clustering results have higher accuracy.

2. Related conception

2.1. DBSCAN algorithm

DBSCAN is a density-based clustering algorithm that finds high-density connected regions, DBSCAN defines a cluster as the maximum set of density connected points, and the DBSCAN algorithm divides the points in the area with enough high density into clusters, which can identify clusters of any shape in the noisy data set.

2.2. Non-IID Learning

Both big data and small data have two problems: one is Heterogeneity, it is reflected in many aspects, such as data type, attribute, data source and data concerned, as well as data mode, structure, distribution and relationship, even learning results may be heterogeneous; The other is coupling relationship, it is not necessarily dependence, association or correlation, but involves many aspects, including complex relationships, levels, types and so on. The two aspects combine is Non-IID.

2.3. Coupling object similarity measurement

2.3.1. Information table

Information table $l=<O, A, V, f>$ organize large numbers of data objects with the same attributes. The information table is shown in Table 1.

| $O$ | $a$ | $a_1$ | $a_2$ | $a_3$ |
|-----|-----|-------|-------|-------|
| $O_1$ | $V_{11}$ | $V_{21}$ | $V_{31}$ |
| $O_2$ | $V_{12}$ | $V_{21}$ | $V_{31}$ |
| $O_3$ | $V_{12}$ | $V_{22}$ | $V_{32}$ |
| $O_4$ | $V_{13}$ | $V_{23}$ | $V_{32}$ |
| $O_5$ | $V_{14}$ | $V_{23}$ | $V_{33}$ |
| $O_6$ | $V_{14}$ | $V_{22}$ | $V_{33}$ |

Where, $O=\{o_1, o_2, \ldots, o_n\}$ consists of a set of finite data objects that are not empty, $A=\{a_1, a_2, \ldots, a_m\}$ is a limited set of attributes, $V=\bigcup_{i=1}^{m} V_i$ represents all attribute values, where, $V_i$ is the attribute value of attribute $a_i$, the information function $f=\mathcal{N}_{i=1}^{m} f_i(O; O \rightarrow V_i)$ assigns specific values for each attribute to each data object.

2.3.2. Coupling object similarity (COS)
Coupling object similarity (COS) can obtain the frequency distribution of feature values and the degree of feature-dependent aggregation with higher accuracy and lower algorithm complexity. Formally, the coupling object similarity between data object \( O_p \) and \( O_q \) is shown in formula (1):

\[
\text{COS} \left( O_p, O_q \right) = \sum_{i=1}^{m} \delta_i \left( O_{pi}, O_{qi} \right)
\]  

(1)

Where, \( O_{pi} \) and \( O_{qi} \) respectively represent the attribute values of data object \( O_p \) and \( O_q \) on attribute column \( i \). \( \delta_i \left( O_{qi}, O_{qi} \right) \) represents the similarity of coupling attribute values between \( O_{pi} \) and \( P_{qi} \) on attribute column \( i \).

2.3.3. Coupling attribute value similarity (CAVS)

Coupling attribute value similarity (CAVS) consists of two parts: Intra-coupled Attribute Value Similarity (IaAVS) and Inter-coupled Attribute Value Similarity (IeAVS). Coupling attribute value similarity between attribute value \( a \) and \( b \) on attribute column \( i \) is shown in formula (2):

\[
\delta_i \left( a, b \right) = \delta_i^{\text{Ia}} \left( a, b \right) \ast \delta_i^{\text{Ie}} \left( a, b \right)
\]  

(2)

Where, \( \delta_i^{\text{Ia}} \left( a, b \right) \) represents the Intra-coupled Attribute Value Similarity; \( \delta_i^{\text{Ie}} \left( a, b \right) \) represents the Inter-coupled Attribute Value Similarity.

2.3.4. Intra-coupled attribute value similarity (IaAVS)

Intra-coupled Attribute Value Similarity (IaAVS) measures the similarity of attribute values by considering the frequency of attribute values under the same attribute. The similarity between attribute values can be characterized effectively from the perspective of frequency distribution. Intra-coupled Attribute Value Similarity is shown in formula (3):

\[
\delta_i^{\text{Ia}} \left( a, b \right) = \frac{\left| g_i \left( a \right) \right| \left| g_i \left( b \right) \right|}{\left| g_i \left( a \right) \right| + \left| g_i \left( b \right) \right| + \left| g_i \left( a \right) \right| - \left| g_i \left( b \right) \right|}
\]  

(3)

Where, \( \left| g_i \left( a \right) \right| \) represents the number of data object \( O_i \) that the attribute value \( a \) appears in the attribute column \( i \). \( \left| g_i \left( b \right) \right| \) represents the number of data objects \( O_i \) that the attribute value \( b \) appears in the attribute column \( i \). \( 1 \leq \left| g_i \left( a \right) \right|, \left| g_i \left( b \right) \right| \leq n \), then, \( \delta_i^{\text{Ia}} \in \left[ 1/3, n/(n + 2) \right] \).

2.3.5. Inter-coupled attribute value similarity (IeAVS)

Inter-coupled Attribute Value Similarity (IeAVS) considers the feature-dependent aggregation degree between attribute values \( a \) and \( b \) in the attribute column \( i \). In other words, it is a comprehensive consideration of the distribution of other attribute column \( k \) \( (k \neq i) \) attribute values under the condition that the attribute values in the attribute column \( i \) are \( a \) and \( b \). Inter-coupled Attribute Value Similarity is shown in formula (4):

\[
\delta_i^{\text{Ie}} \left( a, b \right) = \sum_{l=1,l \neq i}^{m} \alpha_l \delta_i \left( a, b \right)
\]  

(4)

Where, \( \alpha_l \) is the weight parameter of attribute \( a \), \( \sum_{l=1,l \neq i}^{m} \alpha_l = 1, \alpha_l \in [0,1] \). \( \delta_i \left( a, b \right) \) is the similarity of coupling attribute values between attribute values \( a \) and \( b \) under attribute column \( l \) \( (l \neq i) \). \( \delta_i \left( a, b \right) \) is shown in formula (5):

\[
\delta_i \left( a, b \right) = \min_{w \in \cap} \left\{ P_i \left( \{ w \} | a \right), P_i \left( \{ w \} | b \right) \right\}
\]  

(5)

Where, \( \cap \) represents the intersection of all the values of the attribute values of the attribute column \( l \) in the case where the attribute column \( i \) takes the attribute value \( a \) and all the value sets of the attribute values of the attribute column \( l \) under the condition that the attribute column \( i \) takes the attribute value \( b \). \( P_i \left( \{ w \} | a \right) \) and \( P_i \left( \{ w \} | b \right) \) is information conditional probability. \( P_i \left( \{ w \} | a \right) \) describes
the distribution characteristic of attribute value when the attribute value of attribute column i is a and the value of attribute column l is w. The \( P_i(\{w\}|a) \) is shown in formula (6):

\[
P_i(\{w\}|a) = \frac{|g_i(w) \cap g_i(a)|}{|g_i(a)|} \tag{6}
\]

3. NI-DBSCAN algorithm

3.1. Algorithm Though

NI-DBSCAN algorithm uses \( \text{Eps} \) neighbourhood interval value (the neighborhood interval lower limit value \( \text{Eps}_1 \) and the neighborhood interval upper limit value \( \text{Eps}_2 \)) and threshold \( \text{Minpts} \) parameters to divide the threshold of high-density dataset. The similarity between data objects and attribute values is calculated by the coupling similarity formula under Non-IID. The distribution of coupling similarity is output in the form of matrix. Using the scatter plot to fit the distribution of the coupling similarity, find the coupling similarity demarcation point of the data objects between the cluster and the cluster, and use it as the neighborhood interval lower limit value \( \text{Eps}_1 \). At the same time, find the right k-nearest (k takes the value of Minpts) value of the neighborhood interval lower limit value \( \text{Eps}_1 \) and use the maximum value as the neighborhood interval upper limit value \( \text{Eps}_2 \). Select the Minpts value according to the actual situation.

3.2. Algorithm implementation

The key of NI-DBSCAN algorithm is to calculate the similarity between data objects by using the coupling similarity formula and output it as a matrix. The parameter \( \text{Eps} \) neighborhood interval value was obtained by analyzing the coupling similarity matrix. Using the neighborhood interval value can improve the accuracy of the algorithm.

Soybean-small is a thirty-five-dimensional dataset with 47 data objects in three categories. To facilitate the description of the algorithm, this dataset is taken as an example for specific analysis.

3.2.1. Generate coupling similarity matrix

An n-by-n-order matrix \( R \) is adopted to store the similarity between various data objects. Row i column j element \( r_{ij} \) in the coupling similarity matrix \( R \) represents the coupling similarity between the \( i \)th data object and the \( j \)th data object. \( n \) is the number of data objects. Where \( r_{ij} \) represents the value of coupling similarity between data object \( O_i \) and data object \( O_j \), and \( r_{ij} = r_{ji} \). \( r_{ii} \) represents the value of coupling similarity of data object \( O_i \) itself. Therefore, the coupling similarity matrix \( R \) is a real symmetric matrix. The coupling similarity matrix \( R \) is shown in formula (7):

\[
R = \begin{bmatrix}
    r_{11} & \cdots & r_{1j} & \cdots & r_{1n} \\
    \vdots & \ddots & \vdots & \ddots & \vdots \\
    r_{i1} & \cdots & r_{ij} & \cdots & r_{in} \\
    \vdots & \ddots & \vdots & \ddots & \vdots \\
    r_{n1} & \cdots & r_{nj} & \cdots & r_{nn}
\end{bmatrix} \tag{7}
\]

Using the coupling similarity matrix to store the similarity between data objects can reduce the complexity of the algorithm, which is reflected in the following aspects:

- The first is to calculate the coupling similarity of data objects, which only needs to calculate the coupling similarity between the main diagonal and the data objects above the main diagonal, then, \( R = R + R^T \), then all values in the coupling similarity matrix \( R \) can be obtained.
- Second, when judging whether the data object \( O_k \) is the core object is to calculate whether the number of data objects in the neighborhood interval is greater than \( \text{Minpts} \), just find out the number of data objects in the neighborhood of k row vector \( R[k] \) is \( \text{len}(\text{Eps}_1 \leq R[k] \leq \text{Eps}_2) \),
3.3. Maximum to matrix

The difference between coordinate A1 and Eps uses the similarity of the neighborhood interval lower limit value Eps1. For the distance of the data object Oi, the maximum value of the k-nearest neighbor value of the larger side of the similarity of the neighborhood interval lower limit value Eps1, the distance is calculated. Just arrange the rows of elements in matrix R in ascending order. Then find the k value to the right of the lower bound of the neighborhood interval and use the maximum value as the neighborhood interval upper limit value Eps2.

3.2.2. Selection of the Value of neighbourhood interval (Eps1, Eps2) parameters.

Using density connectivity, NI-DBSCAN algorithm can quickly discover clusters of arbitrary shapes, Eps neighbourhood interval value (Eps1, Eps2) uses its special ring structure to narrow the search scope, data objects with high similarity can be better divided into a cluster and clusters of arbitrary shapes can be better discovered.

1) Selection of the neighborhood interval lower limit value Eps1

A scatter plot, also known as an x-y plot, displays all the data in the form of points in a rectangular coordinate system to show the degree of interaction between variables. If the coupling similarity between data objects is small, it will appear as discrete points distributed randomly. If the coupling similarity between data objects is large, most data points will be relatively dense and present in a certain trend. Points that are further away from the cluster are called outliers.

For soybean data set, the number of data objects is taken as the X-axis, and the value of coupling similarity of data objects is taken as the Y-axis. Six data objects were randomly selected to represent the distribution of their coupling similarity using scatter diagram (see Figure 1). Look at the scatter plot and find the dividing point of the dense area that is, the dividing point of cluster to cluster, the value of coupling similarity of data objects at boundary points is taken as the neighborhood interval lower limit value Eps1. Therefore, the Eps1 value of soybean dataset is 31.3.

![Figure 1. Coupling similarity distribution of data objects](image)

2) Selection of the neighborhood interval upper limit value Eps2

The matrix Rs is obtained by ascending arrangement of each row of elements in the coupling similarity matrix Rs. In matrix Rs, for each uncertain data object Oi, find the k-nearest neighbor distance value to the right of Eps1 (k takes the value of Minpts). The k-nearest neighbor set Nk is obtained and the maximum Nk value is taken as the neighborhood interval upper limit value Eps2. For soybean dataset, the Eps2 value was 31.7502.

3.3. Algorithm description

- Input: N uncertain data object sets O=\{o1, o2, ... , on\};
- Output: C cluster H={H1, H2, ..., Hn};
- Step1: Calculate the coupling similarity of each uncertain data object Oi and get the coupling similarity matrix R;
- Step2: Using statistical knowledge to get the parameter Minpts according to the actual situation, the values of neighborhood interval (Eps1,Eps2) were obtained by analyzing the coupling similarity matrix R;
- Step3: Mark all data objects in data object set O as unaccessed;
- Step4: Take any unaccessed data object O_p determine whether the data object O_p is a core object. If so, set up a new cluster, which is expressed as H_p, and add all data objects in the neighborhood interval to H_p. Otherwise, the O_p is treated as a noise point, the data object O_p and all the data objects in its neighborhood are added to the set of noise points, mark the data object O_p as accessed and add all unaccessed data objects in the neighborhood to the SeedList;
- Step5: Take the first data object O_q in the SeedList, determine whether the data object O_q is a core object, if so, mark all data objects in the neighborhood interval as unaccessed and add them to H_p. Otherwise, the data object O_q is regarded as a noise point, add all data objects marked as unaccessed in data object O_q and its neighborhood interval to the noise concentration, mark data object O_q as accessed and remove the data object O_q from the SeedList. At the same time, all unaccessed data objects in the neighborhood interval are added to the SeedList;
- Step6: If there is still a data object in the SeedList, select the first data object in SeedList and repeat step4;
- Step7: Repeat step4-6 until there is no unaccessed data object in data object set O;
- Step8: Output cluster H_1, H_2, … , H_c.

4. Experiment and Evaluation

4.1. Algorithm evaluation index

In this paper, running time is used to evaluate the efficiency of the algorithm and precision is used to measure the quality of the clustering results. The calculation formula of precision is shown in formula (8):

$$\varphi = \frac{n_c}{n} \times 100\%$$  \hspace{1cm} (8)

Where, n_c represents the number of data objects gathered into the correct cluster, n represents the total number of data objects.

4.2. Verification of algorithm

Experimental environment: Operating system Widows7, software: Matlab2016a, processor: Inter (R) core (TM) i7-6700 CPU, memory:16G. In order to verify the effectiveness and feasibility of the proposed algorithm, three data sets (Balloons, Soybean and Zoo) in the UCI machine learning database are taken as experimental data and tested with algorithm accuracy. Table 2 lists the data sets used in the experiment.

| Data Set     | Data Object | Attributes | Cluster Number |
|--------------|-------------|------------|---------------|
| Yellow-small | 20          | 4          | 2             |
| Soybean-small| 47          | 35         | 3             |
| Zoo          | 101         | 17         | 7             |
| Soybean-large| 307         | 35         | 19            |

Table 2. Experiment dataset

In this paper, four data sets of UCI data sets were used to test and compare the clustering effects of DBSCAN, I-DBSCAN, KANN-DBSCAN and the NI-DBSCAN algorithm in this paper. Due to the inconvenience of multidimensional space drawing, only the form of clustering results is given. The clustering results and parameter settings are shown in Table 3.
Table 3. The clustering results and parameter settings

| Data Set      | Clustering Algorithm | Eps/ (Eps1, Eps2) | Minpts | ϕ     |
|---------------|----------------------|-------------------|--------|-------|
| Yellow-small | I-DBSCAN             | 1                 | 7      | 40%   |
|               | KANN-DBSCAN          | 1                 | 3      | 40%   |
|               | NI-DBSCAN            | (3.2280, 3.2299)  | 2      | 80%   |
| Soybean-small| I-DBSCAN             | 2.4629            | 4      | 57.45%|
|               | KANN-DBSCAN          | 2.46              | 2      | 61.70%|
|               | NI-DBSCAN            | (31.3000,31.7502) | 3      | 63.83%|
| Zoo          | I-DBSCAN             | 0.9604            | 3      | 40.59%|
|               | KANN-DBSCAN          | 1.22              | 5      | 62.38%|
|               | NI-DBSCAN            | (14.5000,15.0342) | 7      | 80.20%|
| Soybean-large| I-DBSCAN             | 2.3392            | 5      | 11.73%|
|               | KANN-DBSCAN          | 3.22              | 9      | 22.48%|
|               | NI-DBSCAN            | (31.6000,31.7916) | 4      | 32.89%|

As can be seen from Table 3, the final clustering results obtained by the improved NI-DBSCAN algorithm are significantly better than DBSCAN, I-DBSCAN, and KANN-DBSCAN. This proves that the improved NI-DBSCAN algorithm is effective and feasible. The main reason is that the coupling similarity formula under Non-IID fully considers the true relationship between data objects. At the same time, using statistical principles to obtain parameter Eps neighborhood interval values, it can well find clusters of arbitrary shape and can better select clusters with high similarity, thus improving the clustering quality.

5. Conclusion

DBSCAN is a classic density-based clustering algorithm that automatically determines the number of clusters and can find clusters of arbitrary shapes. For problems the traditional DBSCAN algorithm assumes that data objects are independent and identically distributed and requires two parameters to be manually input, we propose a Density Based Spatial Clustering of Application with Noise under Non-IID (NI-DBSCAN). By using the coupling similarity formula under Non-IID to calculate the similarity between data objects, using statistical principles to obtain parameter Eps neighborhood interval values and thresholds Minpts. Use neighborhood interval values to better find clusters of arbitrary shape. In this process, the true relationship between the data is fully calculated, and a better clustering effect can be achieved.

References
[1] H. Zhou, Y. Chai Soh, X. Wu, “Integrated analysis of CFD data with K-means clustering algorithm and extreme learning machine for localized HVAC control,” Applied Thermal Engineering, 2015, pp.76.
[2] H. Naganathan, W. O. Chong, X. Chen, “Building energy modeling (BEM) using clustering algorithms and semi-supervised machine learning approaches,” Automation in Construction, 2016, pp.72.
[3] David M. Rocke, J. Dai, “Sampling and Subsampling for Cluster Analysis in Data Mining: With Applications to Sky Survey Data,” Data Mining and Knowledge Discovery, vol. 7, no. 2, 2003.
[4] Seyed Mohammad Seyed Hosseini, Anahita Maleki, Mohammad Reza Gholamian, “Cluster
analysis using data mining approach to develop CRM methodology to assess the customer loyalty,” Expert Systems With Applications, vol. 37, no.7, 2009.

[5] Alfonso Capozzoli, Fiorella Lauro, Imran Khan, “Fault detection analysis using data mining techniques for a cluster of smart office buildings,” Expert Systems With Applications, vol. 42, no.9,2015.

[6] B. Ojeda-Magaña, R. Ruelas, M. A. Corona Nakamura, D. W. Carr Finch, L. Gómez-Barba, “Pattern Recognition in Numerical Data Sets and Color Images through the Typicality Based on the GKPFM Clustering Algorithm,” Mathematical Problems in Engineering, 2013.

[7] K.Y Huang, “The use of a newly developed algorithm of divisive hierarchical clustering for remote sensing image analysis,” International Journal of Remote Sensing, vol.23, no.16, 2002.

[8] Jahirabadkar S, Kulkarni P, “Algorithm to determine ε- distance parameter in density based clustering,” Expert Systems with Applications, vol. 41, no. 6,2014, pp.2939-2946.

[9] Khan M M R, Siddique M, Bakr A, et al, “ ADBSCAN: adaptive density-based spatial clustering of applications with noise for identifying clusters with varying densities, ” arXiv : 1809.06189, 2018.

[10] L. Cao, “Non-iidness learning in behavioral and social data,” The Computer Journal, vol.57, no.9, pp.1358-1370, 2014.

[11] L. Cao, “Coupling learning of complex interactions,” Information Processing and Management, vol.51, no.2, pp.167-186, 2015.

[12] C. Wang, X. Dong, F. Zhou, L. Cao, C. Chi, “Coupled attribute similarity learning on categorical data,” IEEE Trans. Neural Netw. Learning Syst, vol.26, no.4, pp.781–797, 2015.

[13] C. Wang , Z. She, L. Cao L, “Coupled attribute analysis on numerical data,” in IJCAI 2013. AAAI Press, Beijing, 2013, pp. 1736-1742.

[14] Z. Xu, Y. Zhang, L Cao, “Social image analysis from a non-IID perspective,” IEEE Transactions on Multimedia, vol.16, no.7, pp.1986-1998, 2014.

[15] G. Pang, L. Cao, L. Cheny, H. Liu, “Learning Homophily Couplings from Non-IID Data for Joint Feature Selection and Noise-Resilient Outlier Detection,” in IJCAI 2017, 2017.

[16] G. Pang, L. Cao, L. Chen, “Outlier detection incomplex categorical data by modelling the feature value couplings,” in IJCAI 2016, 2016, pp. 1–7.

[17] L. Cao, “Non-iid recommender systems: A review and framework of recommendation paradigm shifting,” Engineering, vol.2, no.2, pp.212-224, 2016.

[18] F. Li, G. Xu, L. Cao, “Coupled matrix factorization within non-iid context,” in PAKDD 2015, New York, Springer, 2015, pp. 707-719.

[19] C. Liu, L. Cao, P Yu, “Coupled fuzzy k-nearest neighbors classification of imbalanced non-IID categorical data,” in IJCNN 2014. IEEE. Beijing, 2014, pp.1122-1129.

[20] C. Liu, L. Cao , “A Coupled k-Nearest Neighbor Algorithm for Multi-label Classification,” in PAKDD 2015, Springer, New York, 2015, pp. 176-187.

[21] C. Liu, L. Cao, P. Yu, “A hybrid coupled k-nearest neighbor algorithm on imbalance data” in IJCNN 2014. IEEE. Beijing, 2014, pp. 2011-2018.