Digital Communications and Networks 2 (2016) 145–150

HOSTED BY

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.elsevier.com/locate/dcan

Resilient parallel similarity-based reasoning for classifying heterogeneous medical cases in MapReduce

Haiyan Yu [a,d,e], Jiang Shen [b], Man Xu [c,*]

[a] School of Economics and Management, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[b] College of Management and Economics, Tianjin University, Tianjin 300072, China
[c] Business School, Nankai University, Tianjin 300711, China
[d] School of Computer Science and Engineering, University of Electronic Science and Technology of China, 611731, China
[e] Big Data Research Center, University of Electronic Science and Technology of China, 611731, China

ARTICLE INFO

Article history:
Received 22 February 2016
Received in revised form
21 July 2016
Accepted 29 July 2016
Available online 12 August 2016

ABSTRACT

Given the exponentially increasing volume of heterogenous medical cases, it is difficult to efficiently perform similarity-based reasoning (SBR) on a centralized machine. In this paper, we investigate how to perform SBR using MapReduce (SBRMR), which is an inference framework for data-intensive applications over clusters of computers. To combine the similarities from the individual machines, a mixed integer optimization problem is formulated to filter the priority reference cases. Besides, a resilient mapping mechanism is employed using a quadratic optimization model for weighting the attributes and making the neighborhoods in the same class compact, hence improving the inference capacity. Our experiments on classifying the medical cases demonstrate that SBRMR has approximately 4.1% improvement in classification accuracy over SBR, which suggests that SBRMR is an efficient and resilient similarity-based inference approach.

1. Introduction

Information and Communication Technologies (ICT) provide various solutions to improve the service quality of the health care systems [1] and reduce the rate of misdiagnosis. As a novel application area, medical case classification has attracted more and more attention from the researchers in Clinical Decision Support (CDS) [2,3] due to the huge amount of Electronic Medical Records (EMR) and other heterogeneous patient data available among care providers and healthcare facilities. Many useful patterns on best clinical decision practice could be derived from mining these medical case repositories. The clinical data in health care systems are prone to heterogeneity data, yet each entity data contains different sets of multidimensional attributes [4]. With the multi-dimensional data, assessing similarity/distance between pairwise medical cases is one of the fundamental problems in CDS.

A key motivation is to leverage the concept of inter-entity similarity for case retrievals, knowledge discovery analysis and inferences, and patient cohort identification. Similarity-Based Reasoning (SBR) algorithms have been widely applied in such applications, including information retrieval, discrimination analysis [5–7], etc. The cognitive conception of inter-entity similarity and the related theoretical finding that similar causes bring about similar effects provides the logical foundation of many formal methods, i.e., inductive reasoning [8]. A typical example is Case-Based Reasoning (CBR) [9,10], a problem solving methodology declares that "similar cases have similar solutions". For each observation of medical cases, the decision-maker (i.e., physicians) predicts a set of class labels expressing their beliefs about the underlying probability distribution [4]. However, the previous approaches focusing on these methods are designed to be executed on a single thread on a single machine. This will bias the inference knowledge on the genuine underlying clinical similarity between pairwise entities.

With the exponential increase in the scale of the input datasets [11–13], processing large data in parallel and in a distributed fashion is becoming a popular practice. For this propose, MapReduce [14] is a programming framework for processing a high volume of case datasets by exploiting the parallelism among a cluster of computing nodes. In recent years, MapReduce has gained a lot

* Corresponding author.
E-mail addresses: yhy188@tju.edu.cn (H. Yu), motoshen@163.com (J. Shen), td_xuman@nankai.edu.cn (M. Xu).
Peer review under responsibility of Chongqing University of Posts and Telecommunications.

http://dx.doi.org/10.1016/j.dcan.2016.07.003

of popularity for its flexibility, simplicity and scalability. MapReduce is now well investigated [15] and widely adapted in both scientific and commercial applications. Therefore, MapReduce provides an ideal framework for processing SBR operations over an exponentially increasing volume of medical case data.

Given a medical query, it is very important to convert the beliefs of the decision makers into class labels according to the underlying query data similarities by incorporating the results from multiple machines. To address this problem, in this paper we propose a method to integrate SBR results (similarity metrics with a ranked list of relevant medical cases) obtained from individual machines into a single combined target metric reflecting the true underlying data cases. There are a number of interesting and challenging issues associated with realizing this idea in MapReduce, e.g., how to distinguish similar patterns efficiently in MapReduce, how to reduce the amount of communication and improve speed in the Map-to-Reduce phase. We address these problems in our study.

We design an effective mapping mechanism that exploits combination rules for similar case integration, these both reduce the inference cost and improves the discrimination ability. To reduce the inference cost, we propose a optimization algorithm to select similar neighbors. To improve the discrimination ability, we propose an approximation algorithm to estimate the weights of the attributes. Extensive experiments on our in-house cluster demonstrate that our proposed methods are efficient, robust and scalable. This paper is organized as follows. Section 2 discusses baseline methods and Section 3 presents our parallel SBR algorithms in MapReduce. Section 4 reports our experimental results. Section 5 is our conclusion.

2. Preliminaries

2.1. Overview of SBR

We assume a binary classification task (i.e. diagnose a disease) and a dataset U of historical cases indexed by (i), and P_j^U is the jth partition of U. Each case consists of input values and a class label C, which refers to the physical features of the decision object associated with the task. Here two elements $C_0^{(i)}, C_1^{(i)} \in C$ are used to represent the binary potential outcomes, absent or present, respectively. For query problems, their attribute values are denoted as an input matrix Q, $Q \in \mathbb{R}^d$. When Q contains T entities, the information of each entity is denoted by a vector q, $Q = [q^{(t)}]_{t=1}^{T} = [q_j^{(t)}]_{t=1,j=1}^{T,d}$, where t is the index of the entities. The indicator $\delta^{(i)} \in \{0, 1\}$ refers to the assignment of ith case in U to one query. These selected cases consist of the priority queue P^U. The goal is to find a reference model that predicts the class labels for these queries.

According the theory of similarity-based reasoning (SBR) [8,16], the query can be identified through the nearest neighbor rule and the priority queue P^U is obtained from U. $SBR(q^{(t)}, U) = (P^U, C^q, B^q)$, where C^q and B^q are the class label and its corresponding belief. The similarity metrics have a number of forms, commonly including Mahalanobis distance-based similarity and exponential similarity [17]. The exponential similarity of $SBR(q^{(t)}, U)$ is formulated as:

$$s_w^{(t,i)} = \exp\left(- \sum_{j=1}^{d} w_j (q_j^{(t)} - u_j^{(i)})^2 \right) \tag{1}$$

where $u_j^{(i)}$ and $q_j^{(t)}$ are the jth vector elements from the previous entity data $u^{(i)}$ in U and query information $q^{(t)}$. $w = \{w_j | j = 1, \ldots, d\}$ is an unknown d dimensional vector. The weight w in model SBR can be obtained by domain experts or by a supervised learning method, i.e., mutual information based feature selection [18,19]. The function $s_w^{(t,i)}$ is a real value non-negative similarity function, which satisfies the symmetry, transitive and multiplicative rules.

Algorithm 1 shows the details on how to obtain the class label C^q and its B^q. We first create a priority queue P^U with size m (line 5). For U, we compute $s_w^{(t,i)}$ for each pair of $q^{(t)}$ and $u^{(i)}$. The similarity measure $s(q^{(t)}, P^U)$ is maintained in P^U. To speed up the computation of B^q, we maintain $s_w^{(t,i)}$ in P^U based on the ascending order. Hence, when $s_w^{(t,i)} < P^U.top$, we can guarantee that no remaining objects in U help refine B^q (line 8). Finally, we return the top of P^U which is taken as B^q (line 9).

Algorithm 1. SBR.

1: i: a list of entities;
2: $u^{(i)}$: the ith entity in the case dataset U;
3: $s_w^{(t,i)}$: a similarity function with regard to $u^{(i)}$ and $q^{(t)}$, assuming the weights w given by experts;
4: B^q: belief to be predicted for the class label C^q;
5: P^U: queue of length m whose members are all initialized to $+\infty$;
6: **while** $u^{(i)} \in U$ **do**
7: $sim = s_w^{(t,i)}$;
8: **if** $sim > s(q^{(t)}, P^U)$ **then**
9: $INSERT(u^{(i)}, P^U)$;
10: **end if**
11: **end while**
12: **return** $R = (q, P^U, C^q, B^q)$;

2.2. MapReduce framework

MapReduce [14] is a wide-accepted programming framework to provide data-intensive applications using shared-nothing computing clusters. In MapReduce, input data are reconstructed as key-value pair instances. In its functional programming primitives, Map and Reduce functions are two critical components to process the data. Map function takes a key-value pair as input and generates a series of intermediate key-value pairs. Then, the runtime system groups and sorts all the intermediate values associated with the same intermediate key. Subsequently, Reduce function receives these intermediate keys and their corresponding values. Finally, Reduce function also adapts the processing logic and generates the final result, typically returning a list of values.

2.3. Model weight learning

To obtain the intraclass and interclass similarity matrices, the neighborhoods are divided into two types [20] due to the types of the real class of the queries. For $q^{(t)}$, one type is homogeneous neighborhood, $u^{(i_o)}$, which is the multiple nearest data points of $q^{(t)}$ with the same label. The other type is heterogeneous neighborhood, $u^{(i_e)}$, which is the multiple nearest data points of $q^{(t)}$ with the different label.

Assuming weight $w \in \mathbb{R}^{d \times d}$, w is a Positive Semi-Definite Symmetric (SPSD) matrix. Because there are many features with little use for identifying certain queries, Cholesky decomposition theorem is adopted to achieve $w = WW^T$. With matrix analysis, the similarity matrix (1) is transferred to: $-\ln(s_w^{(t,i)}) = (u^{(i)} - q^{(t)})^T w^T (u^{(i)} - q^{(t)})$, where $\ln(\cdot)$ denotes the natural logarithm function and w^T is the transpose vector of w. Therefore, the optimal problem with the similarity metric can be rewritten as the following discrimination criterion [21]:

$$\min_{W: w^T w = I} \operatorname{tr}(W^T (\Sigma_C^q - \Sigma_S^q) W) \tag{2}$$

where $\mathrm{tr}(\cdot)$ is the trace of the matrix, the integrated matrix $\Sigma_C^q = W^T(u^{(i_e)} - q^{(t)})^T(u^{(i_e)} - q^{(t)})W$,
$\Sigma_C^q = W^T(u^{(i_o)} - q^{(t)})^T(u^{(i_o)} - q^{(t)})W$.

This criterion makes the data instances in the same class compact while data instances in different class diverse locally. Moreover, the orthogonality constraint $w^Tw = I$ $(w^T = W^TW)$ is imposed to eliminate the feature redundancy among different dimensions of W and avoid some arbitrary scaling by controlling the scale of W. The weights of the proposed method have been obtained by the discrimination criterion, which was been verified effectively in [21]. The process of solving this optimization is beyond the scope of this paper, which will present the parallel SBR algorithms in MapReduce.

3. Handling SBR using MapReduce

For large volume dataset, their storage and transfer costs of are much higher than those of the algorithm codes and the query data. Therefore, we design to send the code of SBR and the query data to the multiple data sources, run the algorithm parallely to obtain the results from individual machines, and then combine the results in the decision fusion center.

To see this, we illustrate the parallel SBR framework as shown in Fig. 1. The distributed databases can be cast as cases provided by d sources, $d \geq 1$, and there are d machines involved in this metric integration process. The weight vector w denotes as the importance of each data source. The data characterized by the distributed sources are shown as white circles. The solid black circles depict attribute values of the query, whereas the large circles around the solid black circles depict the spatial uncertainty of the individual data source. After conducting SBR in each machine, the results of them will be transfer to the fusion center (reducer) through the similarity matrix. Then, the conclusion will be derived by combining the results.

In MapReduce, machine $j(1 \leq j \leq d)$ has its own entity feature matrix P_j^U. There is also an associated individual metric $s_w^{(t,i)}$ for machine j. Both P_l^U and $s_w^{(t,i)}$ are separated and hidden from other machines. Each machine will derive and share neighborhood information encoded in two $d \times d$ matrices Σ_C^q and Σ_S^q. The SBR in MapReduce (SBRMR) algorithm is applied through an alternative optimization on all the neighborhood matrices in order to learn a global consistent similarity metric. For all the machines, the only assumption to enable SBRMR is that they share the consistent set of feature dimensions and output label.

The objective of SBRMR is to integrate individual metrics from multiple machines into a global consistent metric. For each machine, SBRMR only requires it to provide the neighborhood information to share with others. The whole process of SBRMR is illustrated in Fig. 2.

3.1. Data preprocessing

A good partitioning of Q for optimizing SBR should cluster objects based on their proximity. There are many data partitioning techniques, i.e., the Voronoi diagram-based data partitioning [22], which are efficient for maintaining data proximity, especially for data in multi-dimensional space. Here, before launching the MapReduce jobs, a preprocessing step is invoked in a master node (the fusion center) with medical domain knowledge. In particular, the data are distributed to d machines for their sources.

3.2. First MapReduce job

The first MapReduce contains a single Map stage, which takes the query dataset and historical dataset Q and U as the input. The result of the mapping stage is a partitioning on U, $\{U_j | j = 1, 2, \ldots, d\}$. Here U_j not only denotes one-dimensional data (i.e, acquired by a single medical sensor), but also multi-dimensional data. Meanwhile, the mappers also collect some statistical information about each partition U_j.

Specifically, before launching the map function, the selected queries Q are loaded into main memory in each mapper. A mapper sequentially reads each entity q from the input split, computes the similarity between q and all reference cases in U_j, and assigns q to the closest cases P_j^U. Finally, the mapper outputs each entity q along with its partition id, original dataset name (U_j), similar to the closest case.

3.3. Second MapReduce job

According to the statistics collected in the first MapReduce job, given the jth replicas Q_j on Q, mappers of the second MapReduce job find the subset U_j of U for each set Q_j . Finally, each reducer performs the SBR operation between a pair of U_j and Q_j received from the mappers.

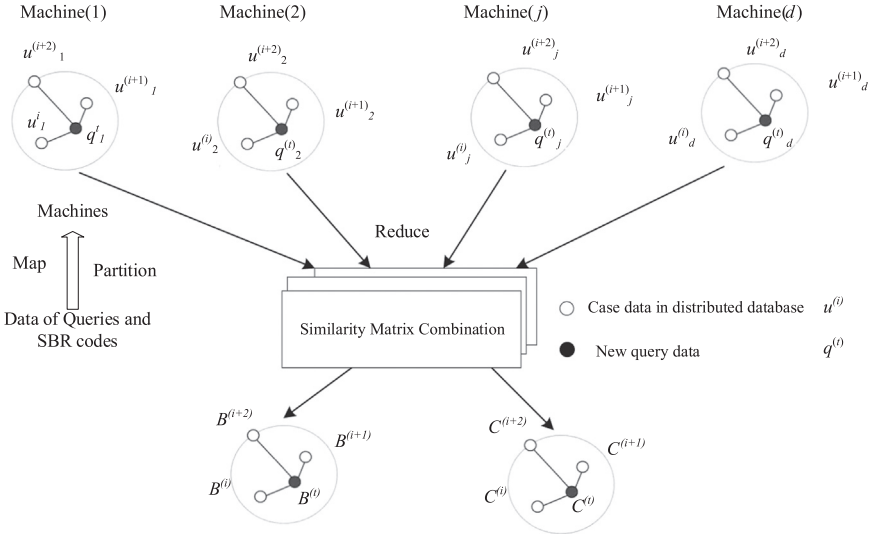

Fig. 1. Illustration of parallel SBR framework.

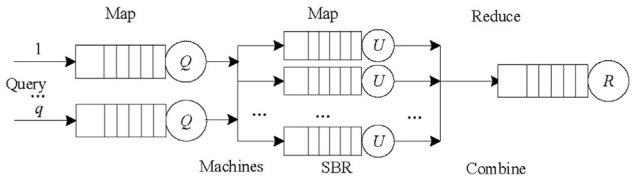

Fig. 2. SBRMR model: parallel SBR in MapReduce.

3.3.1. Mixed integer optimization in a single machine

Each machine identify the true class of all the samples during the training phase. Use SBR to evaluate the accuracy of classifying each of the training samples. To select the optimal subset of neighborhoods, the classification accuracy is maximized. An accuracy matrix $n \times T$ is used as the input of a machine, where T is the size of queries, n is the entity number for case information. Elements in the matrix indicate whether the entity data q is correctly classified with the ith case in the historical dataset or not. The objective of SBR in each machine is to maximize the number of entities correctly classified. For classification decision, $\delta^{(i,q)}$ is the decision variable that the ith case can be selected by SBR model from the historical data, y_q, is a decision variable that indicates whether the query q can be correctly classified. The value elements of the correct classification are the reasoning results of the mixed matrix. With the mixed integer optimization method [23], the selected SBR is given by

$$SR:\ \max \sum_{q\in Q} y_q \tag{3}$$

$$\text{s.t}\begin{cases} \sum_{i=1}^{n} z_i\delta^{(i,q)} - \sum_{i=1}^{n}\frac{1}{2}\delta^{(i,q)} \le My_q, & (4)\\ \sum_{i=1}^{n}\frac{1}{2}\delta^{(i,q)} - \sum_{i=1}^{n} z_i\delta^{(i,q)} + \phi \le M(1-y_q), & (5)\\ 1 \le \sum_{i=1}^{n} z_i \le M, & (6)\\ z\in\{0,1\}^n, y\in\{0,1\}^T, & (7)\\ i=1,2,\ldots,n;\ q\in Q. & (8)\end{cases}$$

where $\delta^{(i,q)}$ denote as the element of the input information matrix; n is the size of the samples in the historical dataset; Q is the query set. $M = n/2$ for no practical significance; and $0 < \phi < 1/2$ is used for identifying the labels of the queries with the neighborhood of cases. The selected neighbor neighborhood improves the quality of the solution and makes the model resilient, because each machine in SBRMR can find their neighborhoods adaptively while the traditional nearest neighbor rule and its inherited method (i.e., Hz-kNNJ [22]) only use the k neighbors for all the queries.

3.3.2. Belief integration

The reducer combines the collected neighborhoods from each machine, then draw the conclusion of the queries with their distribution characteristics. Since $B^{(i)}$ is the real belief of the reference case $u^{(i)}$ in the historical data on class label C_1. For a new query with conditions $q^{(t)}$, $B^{(q)}$ is the belief $q^{(t)}$ on its corresponding class label C_1:

$$B^{(q)} = \frac{\sum_i s_w^{(t,i)}\cdot z_i\cdot B^{(i)}}{\sum_i s_w^{(t,i)}\cdot z_i} \tag{9}$$

where $B^{(q)}$ is the integrated belief with the reference evidence $(u^{(i)}, B^{(i)})$; $s_w^{(t,i)}$ is the similarity matrix between query $q^{(t)}$ and the reference data $u^{(i)}$; z_i indicates that the ith case is chosen by the integer optimization model in each machine.

3.4. SBR combination between Q_j and U_j

As a summary, Algorithm 2 demonstrates the details of SBR procedure that is illustrated in the second MapReduce job. Before launching Map stage, we first partition Q_j for every query set Q (lines 1 and 2). For each entity $q_j^{(t)} \in Q_j$, Map generates a new key value pair, where the key is its partition *id* and the value contains k_1 and v_1 (lines 3 and 4). For each case $u \in U_j$, Map produces a series of new key value pairs, if not pruned based on Eqs. (1) and (2) (lines 7–11). Subsequently, queries in each Q_j and their potential m priority reference cases will be sent to the same reducer. Through parsing the key value pair (k_2, v_2), the reducer derives the query $q_j^{(t)}$ and subset P_j^U that contains $P_1^U, P_2^U, \ldots, P_m^U$ (line 15), and achieve SBR of queries in set Q_j (lines 18–27). For each $q_j^{(t)} \in Q_j$, in order to reduce the complexity of similarity computations, we first rank the partitions from U_j by their similarities in the descending order (line 16). Then, we derive a tighter SBR for every query $q_j^{(t)}$, achieving a higher pruning power. With Eq. (9), we can derive a bound of SBR, B_t, for $q_j^{(t)}$. Hence, we launch a range search with query $q_j^{(t)}$ and threshold B_t [24] over dataset U_j. First, $SBR(q_j^{(t)}, U_j)$ is set to be empty (line 19). Subsequently, all partitions P_j^U are checked one by one (lines 20–26). For each set P_j^U, based on (9), if $s(q_j^{(t)}, P_j^U) < sim$, no cases in $U_j - P_j^U$ can help refine $SBR(q_j^{(t)}, U_j)$, and we go to check the next partition directly (lines 21 and 22). Otherwise, $u_j^{(i)} \in U_j - P_j^U$, if $u_j^{(i)}$ cannot be pruned, we proceed to compute the similarity $s(q_j^{(t)}, U_j - P_j^U)$. If $s(q - j^{(t)}, U_j - P_j^U) < sim$, $SBR(q_j^{(t)}, U_j - P_j^U)$ is refined with $u_j^{(i)}$ and B is refined accordingly (lines 24–26). After checking all the sets U_j, the reducer returns $R(q - j^{(t)}, P_j^U, C_q, B_q)$ (line 32).

Algorithm 2. SBRMR.

map-setup:
1: Q: A list of queries;
2: U_j: The jth case database;
map(k1,v1):
3: **if** $k1$.dataset$=Q$ **then**
4: $pid \leftarrow getPartitionID(k1.\ partition)$;
5: $output(pid, (k1, v1))$;
6: **else**
7: $q_j^{(t)} \leftarrow k1.\ partition$;
8: **while** $P_j{}^U$ **do**
9: **if** $s(q_j^{(t)}, P_j^U) \le k1.\ sim$ **then**
10: $output(t, (k1, v1))$;
11: **end if**
12: **end while**
13: **end if**
14: **reduce(k2,v2):**
15: parse $q_j^{(t)}$ and U_j $(P_1^U, P_2^U, \ldots, P_d^U)$ from (k2,v2);
16: sort $P_1^U, P_2^U, \ldots, P_d^U$ based on the descending order of $s_w^{(t,i)}$
17: $INSERT(u^{(i)}, P^U)$;
18: **while** $q_j^{(t)} \in Q_j$ **do**
19: $B \leftarrow B_j$; $SBR(q_j^{(t)}, U_j) \leftarrow \varnothing$;
20: **while** $j \leftarrow 1$ to d **do**
21: **if** P_j^U can be pruned by the selected SBR (3)–(8) **then**
22: continue;
23: **end if**
24: **while** $u^{(i)} \in U_j - P_j^U$ **do**
25: **if** $U_j - P_j^U$ is not pruned by the selected SBR (3)–(8) **then**
26: refine $SBR(q_j^{(t)}, U_j)$ by $s(q^{(t)}, U_j - P_j^U)$;

27: $B \leftarrow B^{(q)}$ in (9);
28: **end if**
29: **end while**
30: **end while**
31: **end while**
32: **return** $R(q - j^{(t)}, P_j^U, C_q, B_q)$.

4. Experimental evaluation

4.1. Testbed and data source

We adopted a heterogeneous cluster consisting of six nodes, and each of them was configured with one 1.86 GHz Dual-Core and 2 GB RAM. Each node runs on CentOS65 with hadoop-2.7. One of the machines was set as the master node and the rest were set as slaves. The 30 GB of hard drive space was allocated to each Hadoop cluster on each slave and each Hadoop daemon was configured with 1 GB memory. A single NameNode and JobTracker run on the master node. One TaskTracker and DataNode daemon run on each slave node. The chunk size of Distributed File System (DFS) is 128 MB.

The experimental data are clipped from Framingham Heart Study (FHS) dataset [25], which formed heterogeneous multi-source decision-making data as sensor-perceived information. Such dataset consists of heterogeneous information, including EHRs of the patients and physical examinations, resting ECG, laboratory test records, etc. FHS data set contains massive cases. Since obtaining the exact SBR results on the large datasets is very expensive, to study this, we randomly select 4240 records from FHS. In the samples, the average age of the recruiters is 50 years old, and 42.9% male. 15.19% of them have high 10 years risk of heart disease. Each of them was identified with a class label by medical experts, using their expertise and the domain knowledge. For binary classification, they were labeled as CHD present or absent, denoted as C1 and C2. FHS dataset has 581 rows with N/A (missing), and the remaining data were adapted as the experimental data.

4.2. Performance evaluation

To verify the effectiveness of SBR, we adapt *Acc* and Running time as the measures for evaluate the accuracy of discrimination and its execution time of the algorithms. In detail, *Acc* [4] is defined as $Acc = (\eta + \psi)/2$, where η characterizes the accuracy of the real cases identified as C_1 and ψ characterizes the accuracy of the real cases identified as C_0. Meanwhile, these results has been compared with those of SBR [17] and Hz-kNNJ (Hadoop based zkNN Join) [22].

Effect of sizes of discriminative neighborhoods and resilience: For our first experiment, we analyze the inference quality of SBRMR. We measure the quality by the accuracy of the results returned by the related models. We plot the average as well as the 5–95% confidence interval for all randomly selected records. All quality-related experiments are conducted on a cluster with 5 slave nodes, and the number of reducers is also 5. To test the influence of the size of discriminative neighborhoods, we use the FHS datasets and gradually increase neighborhoods from 1 to 20 for Hz-kNNJ [2,4,6,8,10] for SBR and [10,20] for SBRMR, respectively. The accuracy and running times of SBRMR, Hz-kNNJ and SBR with varied neighborhood sizes m are shown in Figs. 3(a) and (b).

Fig. 3(a) indicates that SBRMR exhibits excellent discrimination accuracy (with the average *acc* close to 83.17% in all cases and never exceeds 81% even in the worst case). This shows varying the

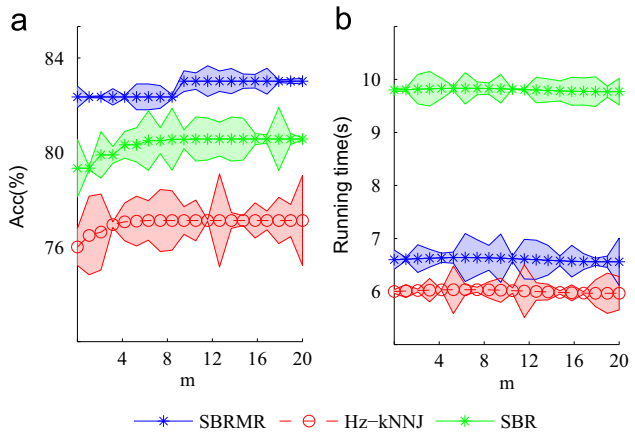

Fig. 3. Accuracy and running time vs. number of neighborhoods m with FHS data.

size of discriminative neighborhoods has almost increase the discrimination accuracy of the algorithm before 2 neighborhoods, and then it tends to converge and stabilize. For *SBRMR*, when the number of neighbors varies from 10 (namely, 2 neighbors/node × 5 nodes) to 20, the accuracy converge to 81.63% even at the worst case. However, for SBR, the accuracy varies with 2-step from 2 to 20, because when the size of the heterogenous neighbors is equal to that of the homogenous neighbors, the accuracy of classifying the medical cases achieves its peak [21]. In detail, when the number of discriminative neighborhoods increases, the accuracy increases linearly due to the larger neighborhood size. For Hz-kNNJ, the tendency of its results is similar to SBR, while the accuracy increases with the size of neighbors changing by 1 step. Fig. 3(b) plots the running time of SBRMR, Hz-kNNJ and SBR when we vary the sizes of discriminative neighborhoods. Clearly, its average running time is near to 9.8 s for SBR all the time, 7.13 s for SBRMR and 6.08 s for Hz-kNNJ. Comparing with SBR and Hz-kNNJ, SBRMR has less variations, which demonstrates SBRMR has resilient capability in discrimination.

Effect of dimension and speedup: We generate $(4240 \times d)$ FHS datasets with dimensionality d from 1 to 15. For these experiments, we also use one more random shift to ensure good reasoning quality results in high dimensions. Fig. 4(a) and (b) present both the accuracy and running time by varying the number of dimensions.

From Fig. 4(a), we observe that the *Acc* of the three approaches follows the similar tendency. In particular, *Acc* increases when n varies from 1 to 6, while it decreases when n varies from 7 to 15. This results from the mutual information of the multidimensional

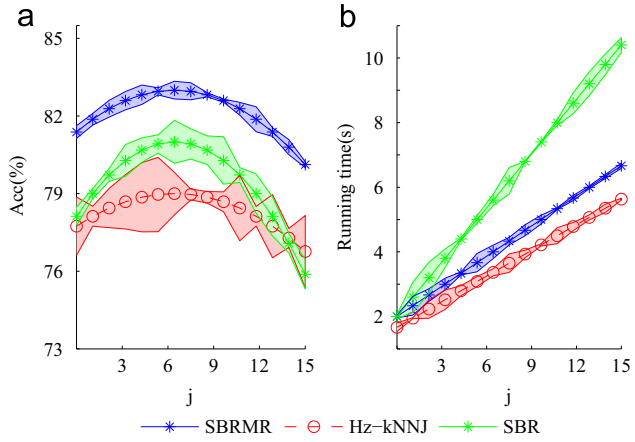

Fig. 4. Accuracy and running time vs. number of dimensions j with FHS data.

features. When the number of dimensions achieves 6 in FHS data, SBRMR achieves its maximum discrimination ability. Comparing with SBR and Hz-kNNJ, SBRMR has 3.25% and 4.1% higher ability in classifying the sample cases, respectively. From Fig. 4(b), the results show that the gap of running time between dimensions j increases. However, the gap of running time between SBRMR and SBR becomes larger when the number of dimensions j increases. The results illustrate that SBRBR is less sensitive to the number of dimensions than SBR, while similar to that of Hz-kNNJ [22]. Based on this trend, it is reasonable to assert that SBRMR outperforms both SBR and Hz-kNNJ, while the improvement in running time is getting less obvious. These results demonstrate that SBRMR has much an efficient and resilient capability in classifying the medical cases.

5. Conclusion

To design and implement similarity-based reasoning (SBR) in a parallel and distributed fashion, we develop the framework of SBR in MapReduce (SBRMR), performing medical case classification over clusters of machines efficiently. SBRMR first construct discriminative neighborhoods from each machine, then it combines all discriminative information in those neighborhoods to learn a single belief matrix. We formulate SBRMR as a mixed integer optimization problem and propose an efficient alternating strategy to filter the priority reference cases. Besides we design an effective mapping mechanism that exploits as a quadratic optimization model for weighting the attributes (or the distributed machines) and making the homogenous neighborhoods compact, and hence improve the inference capacity. Our experiments on classifying medical cases demonstrate that SBRMR has approximately 4.1% and 3.25% improvement in classification accuracy over SBR and Hz-KNNJ, which suggests that SBRMR is an efficient and resilient similarity-based inference approach.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China (71571105, 71601026), NSFC Joint Fund Project of Cross-strait Cooperation in Science and Technology (71661167005) and Science and Technology Research Project of Chongqing Municipal Education Commission (KJ1600401).

References

[1] R. Li, B. Lu, K.D. Mcdonald-Maier, Cognitive assisted living ambient system: a survey, Digit. Commun. Netw. 1 (4) (2015) 229–252.
[2] L.A. Celi, A.J. Zimolzak, D.J. Stone, Dynamic clinical data mining: search engine-based decision support, JMIR Med. Inform. 2 (1) (2014), e13-e.
[3] Man Xu, Jiang Shen, Haiyan Yu, Multimodal data classification using signal quality indices and empirical similarity-based reasoning, Comput. Cardiol. 42 (2015) 1193–1196.
[4] H.Y. Yu, J. Shen, M. Xu, ECs-based reasoning for group decision analysis in the mislabeled classification context, Syst. Eng. Electron. 37 (11) (2015) 2546–2553.
[5] J.P. Brooks, E.K. Lee, Solving a multigroup mixed-integer programming-based constrained discrimination model, INFORMS J. Comput. 26 (3) (2014) 567–585.
[6] J.B. Yang, D.L. Xu, Evidential reasoning rule for evidence combination, Artif. Intell. 205 (205) (2013) 1–29.
[7] L. Fu, M. Shen, H. Yu, et al., Random disturbance reasoning model of decision-making system and its anti-interference capability, Int. J. Ind. Eng. 22 (5) (2015) 645–660.
[8] E. HuLlermeier, Similarity-based inference as evidential reasoning, Int. J. Approx. Reason. 34 (2001) 67–100.
[9] L.-C. Ma, Screening alternatives graphically by an extended case-based distance approach, Omega 40 (2012) 96–103.
[10] Haiyan Yu, Jiang Shen, New algorithm for CBR-RBR fusion with robust thresholds, Chinese J. Mech. Eng. 25 (6) (2012) 1255–1263.
[11] Griffin M. Weber, K.D. Mandl, I.S. Kohane, Finding the missing link for big biomedical data, Jama 311 (24) (2014) 2479–2480.
[12] Christopher C. Yang, P. Veltri, Intelligent healthcare informatics in big data era, Artif. Intell. Med. 65 (2) (2015) 75–77.
[13] J.M. Jordan, Dennis K.J. Lin, Statistics for big data: are statisticians ready for big data?, Int. Chinese Stat. Assoc. Bull. 26 (2014) 59–66.
[14] J. Dean, S. Ghemawat, MapReduce: simplified data processing on large clusters, Proc. Oper. Syst. Des. Implement. (OSDI) 51 (1) (2004) 107–113.
[15] W. Lu, Y. Shen, S. Chen, et al., Efficient processing of k nearest neighbor joins using MapReduce, Proc. VLDB Endow. 5 (10) (2012) 1016–1027.
[16] V. Martinez, G.I. Simari, A. Sliva, et al., CONVEX: similarity-based algorithms for forecasting group behavior, Intell. Syst. IEEE 23 (4) (2008) 51–57.
[17] R.F. Bordley, Using Bayes' rule to update an event's probabilities based on the outcomes of partially similar events, Decis. Anal. 8 (2) (2011) 117–127.
[18] M. Xu, H. Yu, J. Shen, New approach to eliminate structural redundancy in case resource pools using mutual information, J. Syst. Eng. Electron. 24 (4) (2013) 625–633.
[19] H. Yu, J. Shen, M. Xu, Temporal case matching with information value maximization for predicting physiological states, Inf. Sci. (2016), http://dx.doi.org/10.1016/j.ins.2016.05.042.
[20] F. Wang, J. Sun, S. Ebadollahi, Composite distance metric integration by leveraging multiple experts' inputs and its application in patient similarity assessment, Stat. Anal. Data Mining 5 (1) (2012) 54–69.
[21] J. Shen, H.-Y. Yu, M. Xu, Heterogeneous evidence chains based fusion reasoning for multi-attribute group decision making, Acta Autom. Sin. 41 (4) (2015) 832–842.
[22] C. Zhang , F. Li , J. Jestes, Efficient parallel kNN joins for large data in MapReduce, in: International Conference on Extending Database Technology, 2012, pp. 38–49.
[23] W.A. Chaovalitwongse, R.C. Sachdeo, Novel optimization models for abnormal brain activity classification, Oper. Res. 56 (6) (2008) 1450–1460.
[24] S. Alizamir, F. De Vricourt, P. Sun, Diagnostic accuracy under congestion, Manag. Sci. 59 (1) (2013) 157–171.
[25] Lee S. Greta, C. Diane, Y. Qiong, et al., The third generation cohort of the national heart, lung, and blood institute's Framingham heart study: design, recruitment, and initial examination, Am. J. Epidemiol. 165 (11) (2007) 1328–1335.