International Journal of Engineering & Technology, 7 (4) (2018) 5313-5317

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i4.14075
Research paper

Classification Rule Generation for Cancer Prediction using Locality Sensitive Hashing Similarity Measure

Gautam Amiya[1], J Anuradha[2*] and B Venkatesh[3]

[1] *M.Tech, SCOPE Department, Vellore Institute of Technology, Vellore, TamilNadu, India- 632014.*
[2] *Associate Professor, SCOPE Department, Vellore Institute of Technology, Vellore, TamilNadu, India- 632014.*
[3] *Research Scholar, SCOPE Department, Vellore Institute of Technology, Vellore, TamilNadu, India- 632014.*
[*] *Corresponding author E-mail: januradha@vit.ac.in*

Abstract

This paper aims to develop a decision support system for healthcare in predicting stage of cancer (whether benign or malignant) using a novel classifier technique based on Locality Sensitive Hashing (LSH). We propose a new classification rule generations scheme based on Locality Sensitive Hashing. By applying LSH based classification instance selection algorithms, we get a minimal set of class representative patterns, on which we apply discretization and classification rule generation manually .Thus, have high chances of coming up with best prediction. Confusion matrix is used to compare test results. The above technique is applied on two datasets –Iris and Breast Cancer Wisconsin. We get better accuracy, specificity, sensitivity and precision than traditional classifiers. Manual diagnosis takes time and is a trial-error procedure and needs knowledge from medical specialists. We better the accuracy and speed of this manual procedure. classification model concept is used.

Keywords: *CBR (Case Based Reasoning); Discretization; Euclidean Distance Metric; Gaussian distribution; LSH (Locality Sensitive Hashing).*

1. Introduction

Cancer is a widely found illness these days, occurring due to genetically inheritance, environmental contact abuse or accidental mishap. It is of two types –benign and malignant, where malignant is last stage and incurable. By diagnosis of cancer at an early stage can increase the life span of the patient. Different attributes in medical tests are considered to conclude the category of cancer into benign or malignant. This is a very complicated and challenging to arrive to decision by looking into above parameters manually, even by cancer specialist doctors, because most of these parameters are numerical values. We automate this process using our new algorithm based on LSH.

Locality Sensitive Hashing uses the hash functions, hash tables and buckets to place similar objects into same bucket with very high probability and place dissimilar objects in same buckets with very less probability. Thus we form clusters depicted by buckets. First advantage is instead of verifying all the pairs of objects for similarity, we minimize this space and search only among candidate pairs of respective buckets for similar cases. Another advantage of this LSH is that it has linear time complexity. It has some disadvantages also like these buckets are exact and not approximate. Many works have already been done using Case Based Reasoning as a classifier to predict cancer stages. Below are few. Due to open large dataset of breast cancer being accessible which could help in diagnosis, authors proposed schemes to detect cancer stages like Case Based Reasoning (CBR), Rule Based Reasoning (RBR) and Artificial Neural Networks (ANN). In [1] Rossille et al. developed a combination of RBR and CBR scheme using guidelines and cases. CBR has 2 strengths over RBR. a) Cases are better in terms of information transfer and explaining it, b) Majority of areas are complicated and it becomes non-pragmatic to tell all the rules included. So, RBR was joined with CBR, as also it was the first module to be combined with CBR with total success Marling et al. 2002 [2]. Another factor why we join RBR and CBR is that they contribute complementarily. RBR depicts general knowledge and CBR depicts specific knowledge Prentzas et al. [3]. We can also have CBR-first RBR-last way for Medical Diagnosis Decision Support Systems Marling et al.[2]. In [4] tells that CBR is a new way of solving a new problem, a way of adjusting a solution, warn of feasible failures and understanding a situation. Thus main strength is that it is quite novel concept. Another strength is that CBR is an incremental technique.

In [5] developed a medical diagnosis decision support model for gastrointestinal cancer by joining RBR and CBR in CBR-first and RBR-last fashion. It gives as output the probability of a patient having a particular variety of cancer. Also, its accuracy is better than that of only using CBR. As of future work, we should improve following drawbacks. Cross validation technique is not sufficient to produce enough data for a trustworthy model. Similarity measures do not take into account uncertainty, it might wrongly remove some features. Also, we need to gather more No.of cases and use machine learning algorithm on it and methods to fill in missing values.

In [6] developed a CBR system for automatic surveillance and detection of health care related infections. It uses various machine learning tools to (i) Auto gain proof from variety of data including clinical unstructured docs (ii) Involves static and beforehand info used by infection preservationists.(iii) Dynamically produce new information and explanation for each system decision taken. For

bacteria, enteric and urinary infections, good results are achieved in terms of accuracy, Kappa, sensibility and classification. In future we could develop this type of system for other infections apart from bacteria, enteric and urinary infections. In future we should also involve more knowledge having proofs of health care associated infections.

In [7] advices a new strategy applying joining of CBR and clustering to suggest a solution to reduce retrieval space and simplify revise and reuse phases. It uses kNN similarity measure. Its major strength is that it does clustering to reduce search space and then applies CBR. Thus, its results tend to be more accurate and have less time complexity. As Future work rule based reasoning is used for forming clusters and perform comparative analysis.

In [8] studies knowledge-light techniques for adaptation phase of CBR. Two adaptation techniques –adaptation guided retrieval and adaptation on machine learning tools have been developed. Naïve Bayesian classifier and neural networks have been applied. The naive Bayes classifier could adapt the beam number properly only when no change to the beam number of the retrieved case was required. It could be overcome if we take into consideration the interdependence among attributes while finding correct beam number for the new case. The adaptation guided retrieval for beam number had bettered the CBR. For beam angle adaptation we did not get any much satisfactory results showing it is better. As future work adaptation which is more knowledge-intensive and copies the reasoning process are trying to implement.

In [9] developed new technique to depict cases with some flexibility that is required due to complicated cases, various feasible choices and for long durability .The other chief input by the authors is for development of a technique for choosing source cases by utilizing abstraction, conceptualization and inference concepts. Advantages are –it has increased chances of getting a similar case, minimises the time of research, screening valuable properties and weighting them. Its inference techniques lead to better flexibility and creativity of the systems. Disadvantages are that it generates very high number of combinations, random choosing of similar instances leads to loosing few good answers and finally the toughness increases when the state is defined by too many slices. As future work depict knowledge under the state-relation-state structure is going to use. The other improvement is get various states associated with same relation to produce the required elements to get structure made of common definitions.

In [10] joined case based reasoning and multiagent system for using ontology in clinical decision support. In [11] develop a diagnosis decision support system on the basis of logical programming technique to knowledge representation and reasoning using case based reasoning. It is capable of handling explicitly with incomplete, unknown or contradictory knowledge. It is quite accurate in predicting thrombophilia risk. In future, users may assign weights to cases randomly to select most proper plan to solve the problem.

In [12] developed two novel algorithms with linear time complexity for choosing data set instances based on Locality Sensitive Hashing. Its main advantages are that as the traditional schemes take quadratic or log-linear time and cannot handle large volumes of data and lack accuracy, these two algorithms remove these lacunae and are thus better than them. The drawbacks are that these algorithms use very plain techniques to do instance selection and are quite basic. It could be made better by collecting more info regarding instances given to each bucket like incremental average of instances in the bucket or percentage of instances of each class in bucket. Another future work could be production of new instance or a group of these, for each bucket. Another area of future work could be to check if these algorithms are better than other standard algorithms handling streaming data. Also, these algorithms need to be adjusted to work with Big Data.

In [13] developed a root dependent algorithm to better the screening velocity by applying triangle inequality to reduce the searching. Apart from this, a new technique to choose a best root for larger betterment is given. Its main advantage is that it betters the query speed of Euclidean LSH on large sized datasets. It is also discussed that few things can impact the performance of the above algorithm and a sampling technique is suggested to make the algorithm practically more possible. It gives larger query efficiency in all scenarios. Future work could be to design the above algorithm for indexing apart from Euclidean LSH.

In [14] discuss briefly the theory, implementation, performance in terms of accuracy and speed, and applications like discovering similar pages on websites and obtaining similar image and music, of a randomised algorithm called Locality Sensitive Hashing. Advantage of this algorithm is that it finds exact instances in $O(1)$ time not like traditional techniques which take more time. Further advantage of this algorithm is that it has higher probability that it will get a correct match, i.e., higher accuracy. One more advantage is that also one can lessen the LSH fault because of other sources.

In [15] developed a new variant of Locality Sensitive Hashing called TLSH. They developed methods to assess and compare hash values as well as deliver its open source code. Here TLSH has been applied for coding similarity digests. Advantages of TLSH is that it is better than digest techniques available for knowing analogous docs , specifically where missed detections are important.

In [16] assess many families of space hashing functions with respect to each other in a real time situation ,when looking up for high-dimension scale-invariant feature transform (SIFT) descriptors. It matches random projections, lattice quantizers, k-means and hierarchical k-means and concludes that unstructured quantizer betters the exactness of LSH very much. Two new querying techniques are also proposed here and matched with that for LSH. Its pros and cons are also shown.

In [17] bring forth and discuss a new locality sensitive hashing family. It is applicable for the scenarios when the distance is calculated by *ls* norm for s belonging to closed interval-0 through 2. It discusses about LSH technique Based on s-stable distributions, Approximate nearest neighbour, Exact near neighbour, etc.

In [18] discusses in his book about gives theoretical concept and background about LSH-what is shingling of docs, how to preserve similarity of sets using minhashing and minhash signatures, distance measures ,how to apply LSH on text docs, locality sensitive functions and families and amplifying it, applications of LSH.

In [19] Suggested online production of Locality Sensitive Hash Signatures .They discussed LSH with respect to new crux like online hash function and pooling trick.

2. Methodology Used

Figure:1 represents the Architecture of the Proposed System for Classification Rule Generation using Locality Sensitive Hashing method. We have modified from it the original algorithm .The modification is that after applying LSH and deriving Minimal patterns selected as representative of every class, we have applied Discretization on that set and formed classification rules manually and tested this classification rules on test dataset and then analysed the results what we got. We applied Discretization on this minimal set of instances of each class, as the number of rules would be too less and we could form classification rules from Discretization output manually with ease.

1. **Database:** Training datasets of Iris and Breast Cancer made from UCI repository by randomly selecting 70% from the respective total dataset and remaining 30% is the respective test dataset.
2. **LSH Algorithm:**Apply Locality Sensitive Hashing Algorithm on the training dataset using Euclidean similarity measure.
3. Every time dynamically, (suppose n) Hash buckets are formed after AND-OR construction chaining.

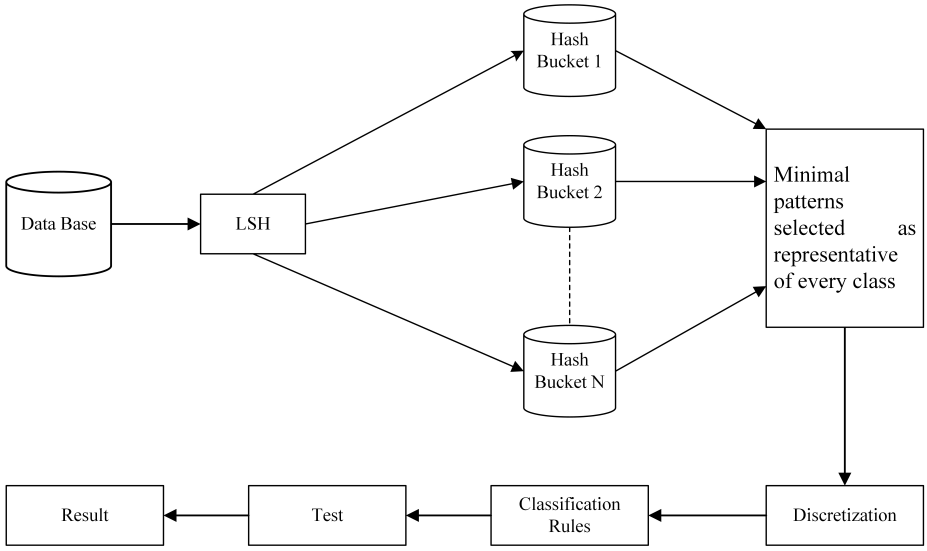

Figure 1: Architecture of Proposed System for Classification Rule Generation for Predicting Cancer using Locality Sensitive Hashing (LSH).

4. From each of these hash buckets formed after AND-OR construction chaining, select one random instance from each of the classes in these hash buckets and add to the set S. This set S thus formed is the minimal set of classification instances selected.
5. Apply Discretization on the above set S that is, replacing continuous features with categorical features.
6. Manually form decision classification rules from the discretization output above obtained.
7. Apply these decision rules on the test dataset and predict the classes.
8. Get the results of the above test and do the analysis how good or bad is the classification rule generation.

Algorithm 1: LSH-IS-F-Classification Rule Generation algorithm by using LSH with two passes.

input : A training set $X = \{(x_1, y_1) \ldots (x_n, y_n)\}$ set of hash function families
output : The set of selected instances $S \subseteq X$ and Discretised Output On that Set

$S = \phi$
foreach *Instance* $x \varepsilon X$ **do**
  foreach *Function family* $g \varepsilon G$ **do**
    $u \leftarrow$Bucket assigned to x by family g
    Add x to u
    foreach *Function family* $g \varepsilon G$ **do**
      foreach *Bucket u of g* **do**
        foreach *Class y with some instance in u* **do**
          $I_y \leftarrow$all instances of class y in u
          if $|I_y| > 1$ **then**
            Add to S one random instance of I_y
          end
        end
      end
    end
  end
end
return S

Load Original Training Data in Excel File Format into Orange Tool. Orange Tool accepts the data domain and the list of data items from Set and returns a new dataset (data sub setting). Call the data processing utility-the data Discretization of the Data Pre-processing Module of Orange Tool. Use the Discretization method Equal Frequency (3 bins with equal no. of data instances) on the above data subset formed. Save the discretized result in a variable and Print it. Form Classification Rules manually from Discretization Output
Instead of one-pass processing, we do more aware choosing of dataset vectors. In first loop, it decides to which bucket identifier each dataset vector corresponds for each AND –construction hash family function and also puts it in that bucket. If only one dataset vector of a class is present, it is not added to the set of chosen dataset vectors .But if more than one dataset vectors of a class exists ,then anyone is randomly selected and put into the set of chosen dataset vector instances. After forming a minimised set of classification instances, apply Discretization on the set so formed by using Orange tool and form classification decision rules manually from the discretised output.

3. Results and Analysis

After running the code, we are getting the dynamic output which is changing every time we run. Below is the output at various stages of this program when we run the code for first time:

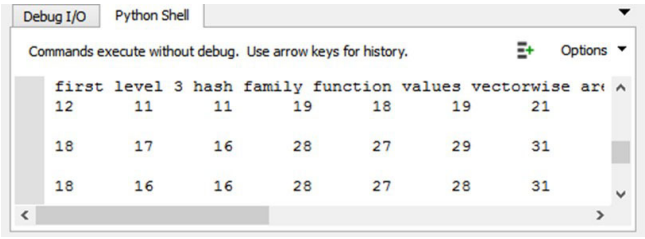

Figure 2: These are the values of 3 hash function family function in the base family vector wise.

These are the values of the 3 basic hash family functions h_1, h_2, h_3 which are generated respectively by putting each of three random vectors generated isotropically and assigning with w=2 and b=1.
These are the three second level Hash Family Functions obtained by concatenating or doing AND-operation on the 3 base family hash functions taking 2 at a time.
These are the set of finally selected data set vectors for classification instance selection. Buckets are the keys in the dictionary and the values are the respective vectors in that bucket.
We aim for Discretization because we are getting a minimal set of classification instances on which we can apply Discretization to get

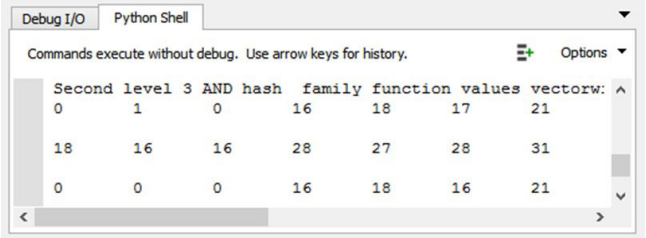

Figure 3: These are the values of 3 second level Hash family

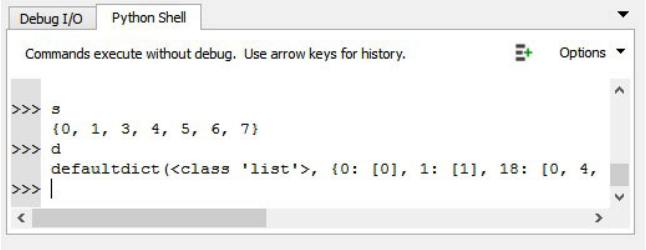

Figure 4: Here set of finally selected data set vectors S and buckets finally created after OR-operation in inner for loop of the algorithm in the dictionary d is shown.

```
Discretised dataset:
[5.15 - 6.25, ≥ 3.25, < 2.45, < 0.75, setosa]
[< 5.15, ≥ 3.25, < 2.45, < 0.75, setosa]
[< 5.15, < 2.95, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[5.15 - 6.25, < 2.95, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[≥ 6.25, < 2.95, ≥ 4.85, ≥ 1.7, virginica]
[5.15 - 6.25, ≥ 3.25, < 2.45, < 0.75, setosa]
[≥ 6.25, ≥ 3.25, ≥ 4.85, ≥ 1.7, virginica]
[< 5.15, ≥ 3.25, < 2.45, < 0.75, setosa]
[≥ 6.25, < 2.95, ≥ 4.85, ≥ 1.7, virginica]
[5.15 - 6.25, 2.95 - 3.25, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[5.15 - 6.25, < 2.95, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[< 5.15, < 2.95, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[< 5.15, < 2.95, < 2.45, < 0.75, setosa]
[≥ 6.25, ≥ 3.25, ≥ 4.85, ≥ 1.7, virginica]
[< 5.15, 2.95 - 3.25, < 2.45, < 0.75, setosa]
[≥ 6.25, 2.95 - 3.25, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[≥ 6.25, < 2.95, ≥ 4.85, ≥ 1.7, virginica]
[≥ 6.25, < 2.95, ≥ 4.85, ≥ 1.7, virginica]
[≥ 6.25, ≥ 3.25, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[5.15 - 6.25, < 2.95, 2.45 - 4.85, ≥ 1.7, virginica]
[≥ 6.25, 2.95 - 3.25, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[< 5.15, ≥ 3.25, < 2.45, < 0.75, setosa]
[5.15 - 6.25, 2.95 - 3.25, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[≥ 6.25, 2.95 - 3.25, ≥ 4.85, ≥ 1.7, virginica]
[≥ 6.25, 2.95 - 3.25, ≥ 4.85, ≥ 1.7, virginica]
[≥ 6.25, ≥ 3.25, ≥ 4.85, ≥ 1.7, virginica]
[< 5.15, 2.95 - 3.25, < 2.45, < 0.75, setosa]
[≥ 6.25, 2.95 - 3.25, ≥ 4.85, ≥ 1.7, virginica]
[≥ 6.25, 2.95 - 3.25, ≥ 4.85, ≥ 1.7, virginica]
[≥ 6.25, < 2.95, ≥ 4.85, ≥ 1.7, virginica]
[5.15 - 6.25, < 2.95, 2.45 - 4.85, 0.75 - 1.7, versicolor]
[< 5.15, 2.95 - 3.25, < 2.45, < 0.75, setosa]
[< 5.15, 2.95 - 3.25, < 2.45, < 0.75, setosa]
[< 5.15, 2.95 - 3.25, < 2.45, < 0.75, setosa]
```

Figure 5: The Set of Discretization Output from the Minimal Set of Selected Instances.

minimal set of classification rules to be formed. Discretization turns continuous numerical values into categorical values.

By increasing the value of random vectors iso-tropically generated, we are able to increase the number of buckets. This is done by selecting the random number generated from Gaussian distribution with mean 0 and increased variance. Also if width of the buckets are increased, there are chances of more number of data instances vectors falling into it, while if the width of the bucket is decreased there are chances of less number of data instances vectors falling into it Also there will be a significant change in the results if we change the value of b as half of w to suppose one-fourth of w. Finally in the Discretization equal frequency methods, the number of bins with equal number of instances should be as many as there are classes in the training data. LSH enhances the efficiency of Nearest Neighbour

Table 1: Effect of Change of Parameter **w**

	Bucket 1	Bucket 2	Bucket 3	Bucket4
#instances when w=2,b=w/2	7	3	7	2
#instances when w=4,b=w/2	9	4	2	-
#instances when w=8,b=w/2	9	6	5	-
#instances when w=16,b=w/2	9	-	-	-

Table 2: Confusion Matrix for Iris Dataset

		PredictedClass		
		Setosa	Versicolor	Virginica
Actual Class	Setosa	**14**	1	1
	Versicolor	1	**11**	3
	Verginica	1	3	**10**

estimation by making it faster. Although the time complexity is unchanged because loop nesting and structure is unchanged, except that KNN stage is bettered. In [20] it has linear time complexity because the instances are chosen into buckets using the singular loop of the dataset by applying the Euclidean Distance hash function on the datasets [21, 22].

In general, the results are giving good accuracy, precision, specificity, sensitivity. Here, are the confusion matrix for the 2 datasets –Iris and Breast Cancer Wisconsin Original, taken from UCI Machine Learning Repository online [23]. Results are quit well as the sensitivity; specificity, precision and accuracy are all greater than 75%.

Table 3: Confusion Matrix for Wisconsin Breast Cancer Predicted Class

		Predicted Class	
		Benign	Malignant
Actual Class	Benign	**113**	29
	Malignant	13	**50**

Table 4: Classifier Results for Iris and Wisconsin Breast Cancer Datasets

	Sensitivity	**Specificity**	**Precision**	**Accuracy**
Iris	77.77	88.87	77.76	85.18
Breast Cancer	79.15	79.24	78.05	79.24

4. Conclusion

This paper has developed completely new way of doing classification prediction, especially for detecting benign and malignant cancer. It has linear time complexity which is better than N Nearest Neighbors Estimation. Also in terms of sensitivity, specificity, precision, accuracy it is decent algorithm. Also the algorithm is tested on large data and found to be suitable for Big Data also. It could be bettered in terms of accuracy, precision, sensitivity, specificity etc. It also needs to be tested on the other data sets of varying dimensionality. We could also include extra knowledge of the instances by putting them in buckets. As Future work we would like to implement the algorithm for parallel streams of vast data for fast execution and more accurate.

References

[1] D. Rossille, J.-F. Laurent, and A. Burgun, "Modelling a decision-support system for oncology using rule-based and case-based reasoning methodologies," *International journal of medical informatics*, vol. 74, no. 2-4, pp. 299–306, 2005.

[2] C. Marling, M. Sqalli, E. Rissland, H. Muñoz-Avila, and D. Aha, "Case-based reasoning integrations," *AI magazine*, vol. 23, no. 1, p. 69, 2002.

[3] J. Prentzas and I. Hatzilygeroudis, "Categorizing approaches combining rule-based and case-based reasoning," *Expert Systems*, vol. 24, no. 2, pp. 97–122, 2007.

[4] J. Kolodner, *Case-based reasoning*. Morgan Kaufmann, 2014.

[5] R. Saraiva, M. Perkusich, L. Silva, H. Almeida, C. Siebra, and A. Perkusich, "Early diagnosis of gastrointestinal cancer by using case-based and rule-based reasoning," *Expert Systems with Applications*, vol. 61, pp. 192–202, 2016.

[6] H. Gómez-Vallejo, B. Uriel-Latorre, M. Sande-Meijide, B. Villamarín-Bello, R. Pavón, F. Fdez-Riverola, and D. Glez-Peña, "A case-based reasoning system for aiding detection and classification of nosocomial infections," *Decision Support Systems*, vol. 84, pp. 104–116, 2016.

[7] A. Mansoul and B. Atmani, "Clustering to enhance case-based reasoning," in *Modelling and Implementation of Complex Systems*. Springer, 2016, pp. 137–151.

[8] S. Petrovic, G. Khussainova, and R. Jagannathan, "Knowledge-light adaptation approaches in case-based reasoning for radiotherapy treatment planning," *Artificial intelligence in medicine*, vol. 68, pp. 17–28, 2016.

[9] P. Chazara, S. Negny, and L. Montastruc, "Flexible knowledge representation and new similarity measure: Application on case based reasoning for waste treatment," *Expert Systems with Applications*, vol. 58, pp. 143–154, 2016.

[10] Y. Shen, J. Colloc, A. Jacquet-Andrieu, and K. Lei, "Emerging medical informatics with case-based reasoning for aiding clinical decision in multi-agent system," *Journal of biomedical informatics*, vol. 56, pp. 307–317, 2015.

[11] J. Vilhena, H. Vicente, M. R. Martins, J. M. Grañeda, F. Caldeira, R. Gusmão, J. Neves, and J. Neves, "A case-based reasoning view of thrombophilia risk," *Journal of biomedical informatics*, vol. 62, pp. 265–275, 2016.

[12] Á. Arnaiz-González, J.-F. Díez-Pastor, J. J. Rodríguez, and C. García-Osorio, "Instance selection of linear complexity for big data," *Knowledge-Based Systems*, vol. 107, pp. 83–95, 2016.

[13] X. Gu, Y. Zhang, L. Zhang, D. Zhang, and J. Li, "An improved method of locality sensitive hashing for indexing large-scale and high-dimensional features," *Signal Processing*, vol. 93, no. 8, pp. 2244–2255, 2013.

[14] M. Slaney and M. Casey, "Locality-sensitive hashing for finding nearest neighbors [lecture notes]," *IEEE Signal processing magazine*, vol. 25, no. 2, pp. 128–131, 2008.

[15] J. Oliver, C. Cheng, and Y. Chen, "Tlsh–a locality sensitive hash," in *Cybercrime and Trustworthy Computing Workshop (CTC), 2013 Fourth*. IEEE, 2013, pp. 7–13.

[16] L. Paulevé, H. Jégou, and L. Amsaleg, "Locality sensitive hashing: A comparison of hash function types and querying mechanisms," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1348–1358, 2010.

[17] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-neighbor methods in learning and vision: theory and practice (neural information processing)*. The MIT press, 2006.

[18] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014.

[19] B. Van Durme and A. Lall, "Online generation of locality sensitive hash signatures," in *Proceedings of the ACL 2010 conference short papers*. Association for Computational Linguistics, 2010, pp. 231–235.

[20] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[21] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 2006, pp. 459–468.

[22] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

[23] K. Bache and M. Lichman, "Uci machine learning repository," 2013.