Journal of Computer & Robotics 14 (1), 2021 57-67

Computer & Robotics

Multi-Instance Learning (MIL) By Finding an Optimal Set of Classification Exemplars (OSCE) Using Linear Programming

Mohammad Khodadadi Azadboni [a], Abolfazl Lakdashti [b],*

[a] *Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic*
[b] *Faculty of Computer Engineering, Rouzbahan University, Sari, Iran*

Received 8 February 2020; accepted 26 May 2021

Abstract

This paper describes how to classify a data set by using an optimum set of exemplars to determine the label of an instance among a set of data for solving a classification run time problem in a large data set. In this paper, these exemplars purposely have been used to classify positive and negative bags in a synthetic data set. There are several methods to implement multi-instance learning (MIL) such as SVM, CNN, and Diverse density. An optimum set of classifier exemplar (OSCE) is used to recognize positive bag (contains tumour patches). The goal of this paper is to find a way to speed up the classifier run time by choosing a set of exemplars. A linear programming approach is been used to optimize a hinge loss cost function, in which estimated label and actual label is used to train the classification. The estimated label is calculated by measuring the Euclidean distance of a query point to all of its k nearest neighbours and an actual label value. To select some exemplars with none zero weights, two solutions are suggested to have a better result. One of them is choosing k closer neighbours. The other one is using LP and thresholding to select some maximum of achieved unknown variable which are more significant in finding a set of exemplars. Also, there is a trade-off between classifier run time and accuracy. In a large data set, the OSCE classifier has better performance than ANN and K-NN cluster. Also, OSCE is faster than the NN classifier. After describing the OSCE method, this has been used to recognize a data set that contains cancer in synthetic data points. Eventually, the defined OSCE has been applied to MIL for cancer detection.

Keywords: (Integer linear programming (ILP), linear programming (LP), exemplar, hinge loss function, Multi-instance learning (MIL), positive bag)

1.Introduction

A significant topic in machine learning is classification, in which learning machines have to know how to group a data set by particular criteria. Here is a supervised learning process where computers group data together based on predetermined characteristics, called supervised classification. Also, there is an unsupervised version of the classification, called clustering when categories are not specified. K-means is an unsupervised learning algorithm used for clustering problem whereas KNN is a supervised learning algorithm used for classification. Classification cannot completely cope with false-negative and positive problems. There are two reasons for these issues: one is related to the complexity of large data set and the other one relates to the classifier method which uses a linear or nonlinear function.

In this paper, we try to cope with both issues by choosing a set of exemplars. In a large data set, because of a huge number of instances in the data set, many math calculations and high execution time of classification are unavoidable. To overcome these issues, choosing a set of the exemplar is suggested to calculate the distance of a query point with

*Corresponding author : E-mail:mohammad64kh@gmail.com

these exemplars instead of all data set to classify the query point. In other words, some instances can be selected to classify the data set. R.Weber in [1] said, for large n, the nearest-neighbor problem is prohibitively expensive since it requires O(n2) distance evaluations. In low dimensions (say d < 10), regular spatial decompositions like quadtrees, octrees, or KD-trees can solve the NNs problem using O(n) distance evaluations [1]. But in higher dimensions, it is known that tree-based algorithms end up having quadratic complexity [2]. To overcome this problem, we must abandon the concept of exact searches and settle for approximate searches. State of the art methods for the NNs problem in high dimensions use randomization methods, for example, tree-based methods [3], [4] or hashing based methods [5]. But in this document, all data points and their neighbors are examined to choose optimum exemplars to classify data queries. Y. Li and X. Zhang [6] proposed k Exemplar-based Nearest Neighbor (kNN) classifier which focused on the two-class imbalanced classification problem, where the majority class is the negative and the minority class is positive. In contrast to most concept learning systems, instance-based learning or k-NN classification, does not conceptual model at the training stage [6].
In addition, an optimization problem is a problem of finding the best solution from all feasible solutions. Optimization problems can be divided into two categories depending on whether the variables are discrete or continuous. Integer linear programming-ILP is known as an optimization problem with discrete variables. In linear programming-LP, problems with continuous variables will be optimized. In a large data set, ILP cannot solve the optimization problem. In our proposed algorithm, both ILP and LP are compared to understand which one can achieve the best set of exemplars and is faster. In large data set we use LP to optimize the weights and then thresholding to select some maximum weight to determine some exemplars.
The purpose of defining the OSCE classifier is to recognize positive bags which contain cancer data set. There are several classifier methods to do this issue.
MIL is strongly studied for more than one decade. Zhou [11] and Zhang [12] respectively in 2002 and 2004 used MIL based on NN. Also, Liu [13] in 2018 and T.Khatibi, A.Shahsavari, A.Farahani [14] in 2021, used deep multi-instance learning based on CNN. In 2020 a SVM based method proposed by X.Wang et. al [15]. There is a trade-off between time consumption and accuracy at MIL proposed methods that use a big data set.

2. Proposed Method

Text should be produced within the dimensions shown on these pages; each column 8.47 cm wide with 0.85 cm middle margin, total width of 17.78 cm and a maximum length of 21cm on the first page and 23.5cm on the second and following pages. Make use of the maximum stipulated length apart from the following two exceptions: (i) do not begin a new section directly at the bottom of a page, but transfer the heading to the top of the next column; (ii) you may exceed the length of the text area by *one line only* in order to complete a section of text or a paragraph.
A thing serving as a typical example or appropriate model is called an exemplar. In an exemplar-based classifier, some training data points select as exemplars to classify the testing data points based on their distance with exemplars. An optimal set of classification exemplars (OSCE) is a classifier based on the optimization loss function of any data points that considering its neighbors labels and weights. This classifier is optimized by LP.

2.1. Overview

The main idea of the proposed optimal set of exemplar comes from the high dimension and dense data set that needs more time to

classify. The accuracy of classification would increase if the training set contains comprehensive data points. At the first step, the proposed algorithm is tested on a big neighborhood and use the euclidean distance to calculate the weight $w_i(\mathbf{x}_q)$for each training points (equation 2). But in the combinatorial optimization method for ILP, if we use a large data set and big neighborhood, not only, the execution time is very high, but the optimization algorithm also may encounter an infeasible solution. To cope with this problem, a small neighborhood around the query point suggested having lower execution time and better decision.

Before explaining the proposed classifier method "an optimal set of classification exemplars (OSCE)", some definition will explain. In the following, the definition of K-NN, kernel-based, and exemplar-based classification will be explained.

2.2. k-NN Classification

For classification data set based on neighborhood, k-nearest neighbors (k-NN) is well known in which finds k nearest neighbors among a query point.

In the below equation, the distance is sorted ascending. And the first k of them are the closest neighbors.

$$\begin{cases} KNN\ (\mathbf{x}_q)=\{\text{SortArray}[n] \mid n=[1 \quad k]\} \\ \text{SortArray}=|\mathbf{x}_i-\mathbf{x}_q| \end{cases} \tag{1}$$

In the above equation SortArray is sorted by following pseudo code:

```
For i = 1 : length(SortArray) - 1
  x = SortArray[i]
  y = SortArray[i+1]
  if x>y
    memory = SortArray[i+1]
    SortArray[i+1] = SortArray[i]
    SortArray[i] = memory
```

2.3. Kernel-Based Classification

The kernel function, a function returning the inner product between mapped data points in a higher-dimensional space, is a foundational building block for kernel-based learning methods. Several linear algorithms can be formulated, whether for clustering, classification, or regression. In Eq.2, w˜i(xq) is the kernel of this equation. kernel methods are suitable for a variety of classification tasks such as [7], [8] . In this document, the weight for estimating the label is the kernel.

$$\tilde{y}_q(\mathbf{x}_q) = \sum_{i \in \mathbf{E}(q)} \tilde{w}_i(\mathbf{x}_q) \cdot y_i \tag{2}$$

$$\tilde{w}_i = \frac{w_i}{\sum_{i \in \mathbf{E}(q)} w_i} \tag{3}$$

$$w_i(\mathbf{x}_q) = \left\|\mathbf{x}_q - \mathbf{x}_{\max}\right\|_2^2 - \left\|\mathbf{x}_q - \mathbf{x}_i\right\|_2^2 \tag{4}$$

$$\mathbf{x}_{\max} \cong \underset{\mathbf{x}_i \in E(q)}{\operatorname{argmax}} \left\|\mathbf{x}_q - \mathbf{x}_i\right\|_2^2 \tag{5}$$

$w_i(\mathbf{x}_q)$ is a positive weight and x_{max} is the furthest point from $\mathbf{x}_q$ in its neighborhood. And $i \in E(q)$ means neighborhood of a query point in the data set $\mathrm{E(q)} \subseteq \Omega$. Also, $\tilde{w}_i$ is a normalized weight.

$\mathrm{E(q)} = \{x_i | x_i \in \mathrm{KNN}(x_q)\}$

2.4. Exemplar-Based Classification

Exemplar-based classification is a subset of classification in which some data points chosen as an important sample of data set to increase the computation time faster. So, Eq. 2 changes to the bellow equation. In this equation η_i is a binary vector of neighborhood $\mathbf{x}_q$, and $\eta_i = 1$ means $\mathbf{x}_q$ chosen as an exemplar of the data set.

$$\tilde{y}_q(\mathbf{x}_q) = \sum_{i \in \mathrm{E}(q)} \eta_i \cdot \tilde{w}_i(\mathbf{x}_q) \cdot y_i \quad , \quad \eta_i \in \{0 \ , \ 1\} \qquad (6)$$

This classification has used since 1980. For instance, Hintzman et. al [9] and Mcandrews et. al [10] used an exemplar- based classifier in the 1980s.
How to implement OSCE in MIL based approach, the patch location is not obvious. We just know the label of the bag. So, for training by OSCE classifier, instead of patch label, which is unknown we use bag label.

2.4. Implementation OSCE

In this subsection, an optimal set of classification exemplars (OSCE) is explained. For speeding up training OSCE classifier run time, combinatorial optimization is used to minimize cost function to choose some exemplars. OSCE identify exemplars among a data set that are an important sample of the data set and use them to reliably derive the optimal subset of instances to classify each query point's label. In this algorithm, the objective function is minimized to find the optimal cost value. Fig. 1 illustrates a set of exemplars in the data points. The following pseudo-code shows the randomly choice of OSCE algorithm:

Ψ = inf, counter = 0

while (counter < 1000)

1) Randomly generate binary η_i for each data points by random permutation function in python that $\sum_{i \in \Omega} \eta_i = M$, Ω is training set, and M is number of exemplars.
Sum=0
for each query points do :
find k-NN (Eq. 1)
calculate distance (Eq. 3)
estimate label $\tilde{y}_q$ (Eq. 6)
cost_function = max(1 - $\tilde{y}_q \cdot y_q$, 0)
sum = sum + cost_function

2) if sum < Ψ
Ψ = sum
counter += 1
end

In OSCE proposed algorithm, finding minimum criterion is proportion to optimum exemplars (η_i). To choose some exemplars, if a data point candidate as an exemplar, its η_i has one value and vice versa. So, integer linear programming (ILP) can be formulated in the form of cost function:

This algorithm takes one variable:

- η_i , $i \in \Omega$ (Ω *is training set*)

and three inputs:

- $\mathbf{x}_i \in \mathrm{R}^d$
- $y_i \in \{-1, 1\}$
- $\mathrm{M} \in \mathrm{N}$

Where η is an unknown weight vector. Also, for each i $^{i \in \Omega}$ (training set), there is a feature vector $^{\vec{X} \in R^d}$ (d is constant) and a known label $^{y_i \in [-1 \ , \ 1]}$. M is maximum number of exemplars.

Objective function:

$$\begin{cases} \min \mathrm{L}, \\ \mathrm{L} = \sum_{q \in \Omega} \psi_{\mathrm{q}}(\tilde{y}_q, y_q) \end{cases} \quad (7)$$

In this equation, $\tilde{y}_q$ is an estimation of a query point label around its neighborhood, and Ψ is a hinge loss function:

$$\begin{cases} \tilde{y}_q(\mathbf{x}_q) = \sum_{i \in \mathrm{E}(q)} \eta_i \cdot \tilde{w}_i(\mathbf{x}_q) \cdot y_i \ , \ \eta_i \in \{0 \ , \ 1 \ \} \\ \psi_q(\tilde{y}_q, y_q) = \max(k - \tilde{y}_q y_q, 0) \ , \mathrm{k} > 0 \end{cases} \quad (8)$$

To optimize the min cost function and find proper exemplars, some constraints considered. In ILP, the unknown variable ηi is a binary variable. Beside, η_i = 1 means $\boldsymbol{x}_\mathrm{i}$ is selected as an exemplar. Since, the number of exemplar is defined to be $M \in Z^+$; So the sum of η_i cannot be bigger than M.

Constraint:

$$\boldsymbol{a)} \quad \sum_{\boldsymbol{i} \in \boldsymbol{\Omega}} \boldsymbol{\eta_i} \leq \mathbf{M}$$

$$\boldsymbol{b)} \quad \boldsymbol{\eta_i} \in \{\mathbf{0} \ , \mathbf{1}\}$$

M is an input to choose some exemplar.The definition of a linear programming is: (max or min cT x , Ax ≤ b).

In linear programming, math operators except plus, minus, and multiply constant cannot be used. Therefore, in Eq. 7, the hinge function changes to the following form to have a linear programming model:

$$\psi_q(\tilde{y}_q, y_q) = \max(k - \tilde{y}_q y_q, 0) \Rightarrow \begin{cases} \psi_q(\tilde{y}_q, y_q) \geq k - \tilde{y}_q y_q \\ \psi_q(\tilde{y}_q, y_q) \geq 0 \end{cases} \quad (9)$$

So, according to the Eq. 9, for implementing with Gurobi library Ψq (hinge loss) is defined as a variable. Therefore, there are two other constraints:

$$c) \ \psi_q(\tilde{y}_q, y_q) \geq 0$$

$$\mathrm{d)} \ \psi_q(\tilde{y}_q, y_q) \geq k - y_q \left[\sum_{i \in E, q \neq i} \eta_i . w_i(\mathbf{x}_q) . y_i \right]$$

In the hinge loss function, $\max(k - \tilde{y}_q y_q, 0)$, there is a penalty with the value of k > 0. Here we set the penalty to one. In practice, it can be any positive number. In the current data set with 0 < k < 1 the same results has been achieved. But for the result section k=1, has been assumed.

After training, a η_i with value one is correspond to a selected exemplar, All exemplars are used to predict the label of a query point xq by the following equation:

$$\bar{y}_q(\mathbf{x}_q) = \mathrm{sgn}(\tilde{y}_q(\mathbf{x}_q)) \quad (10)$$

The OSCE, in which a big neighborhood for calculating weights is used, is tested on some different data set with two groups. According to the results, in a small data set (100 instances) the execution time is fast enough. But in the larger data set (1000 instances) it will take more than three hours to execute. Also, in this case, in very large data sets there is no feasible solution for ILP to solve the problem. The other interesting point is that, if we change the unknown integer variable η to continue form (change ILp to LP), in the same large data set and big neighborhood, the execution time will be very faster (18s). Fig. 2 illustrates a comparison of training classification computation time with the big and smaller neighborhood for ILP and LP in different data size. The training run time for both ILP and LP shows that in the modified neighborhood (the smaller neighborhood in

Fig.2-b) the speed is faster than the previous one. Fig.1 shows the result of our proposed method. In this figure the accuracy is 88 percent for the data set with mean and variance of {mp= [1.8, 1.8], mn= [0, 0]} and {varp= [0.8, 0.8], varn= [0.5, 0.5]} respectively. The low accuracy is because of the big neighborhood and big overlap of the data set. In the other words, if the neighborhood is smaller, the accuracy will increase. In order to solve the classification training run time, two solutions proposed:

Firstly, instead of calculating the weight of each query point with a big neighborhood, find a smaller neighbor- hood. It would result from a flexible solution to optimize η.

Secondly, instead of ILP use the LP and thresholding to the achieved η to choose k maximum number of η. The number of selected η is equal to number of exemplars.

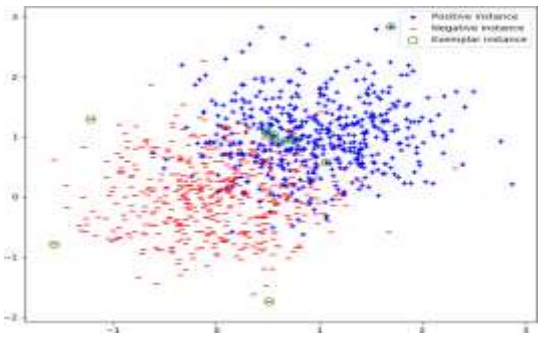

Fig. 1: Set of exemplars at two feature spaces and two classes.

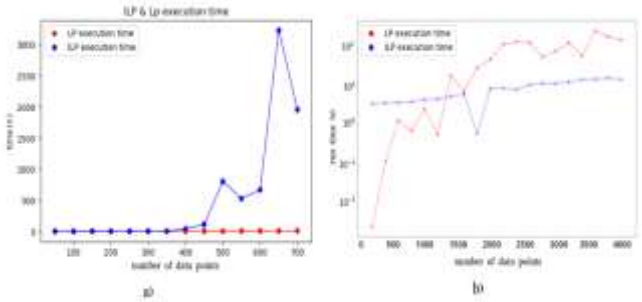

Fig. 2: Comparison of training time for ILP and LP with different data set size and two groups. a) big neighborhood, b) small neighborhood for each query point.

3. Implementation Mil By Osce

- Creating data set Two sets of normal (Fig. 3) and tumour (Fig. 4) bags are considered as negative and positive bags combine these two sets to create a complete data set.
- Thresholding Figure 5 indicates the classifier output in 100000 data points. As it shows there is a large overlap between positive and negative bag. All the negative data points are overlapping. So, a set of Exemplar should be in the positive data points. After finding some exemplars, to classify data points a threshold used manually to find the decision boundary. Indeed, in order to predict the label of the bag, find k closer data points of the bag to the exemplars. In positive bags, the majority of the positive data points are more; So, there is a shorter distance of data points to the exemplars (see fig. 5).

$$D(\mathbf{x}_e) = |\mathbf{x}_e - \mathbf{x}| \tag{11}$$

In above equation **D** is sorted and calls it $\mathbf{D}_{sort}$, then sum the k=5 first closest member of it:

$$D_{closer}(\mathbf{x}_e) = \{D_{sort}(n) \mid n = [1\ k]\} \tag{12}$$

$$\tilde{y} = \sum D_{closer}(\mathbf{x}_e) * \mathbf{y}_e \tag{13}$$

$\mathbf{y}_e$ is the label of the exemplars and $\tilde{y}_e$ is the estimated label of the bag. Also, $\bar{y}$ is the predicted label of the bag:

$$\bar{y} = \begin{cases} 1, & \tilde{y} \prec \text{Thr} \\ 0, & \text{O.W} \end{cases} \quad (14)$$

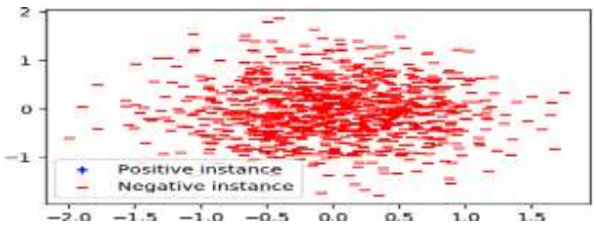

Fig. 3: A normal bag of 1000 patches and 2 features (2000 data points).

4. Experiments

In order to evaluate the proposed algorithm, we test it on different data set with different size (50 to 1000 data points). In each one of the data set, there are two groups of positive and negative instances which each of them have 2 feature vectors with mean and variance of $\{\vec{m\_p} = [2,2],\ \vec{m\_n} = [0,0]\}$ and $\{\vec{\text{var}\_p} = [0.6,0.6],\ \vec{\text{var}\_n} = [0.5,0.5]\}$ respectively.

The following figures show execution time comparison of the for ILP and LP, accuracy and criterion. ILP in comparison with Lp is not very flexible to optimize the solution. Also, in a large data set the difference between execution time is clearly very high. To solve the complexity of time, we use the LP algorithm and thresholding to achieve unknown variable η. In a big neighborhood and using LP, the accuracy is better than ILP. Fig. 6 shows the true-positive, true-negative, false- positive, false-negative, and a set of exemplars.

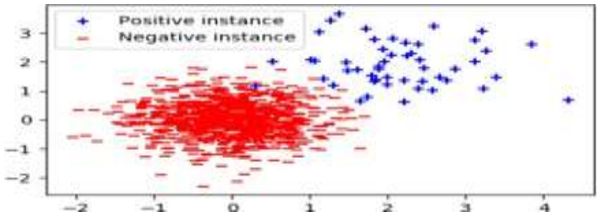

Fig. 4: A tumor bag of 1000 patches and 2 features (2000data points).

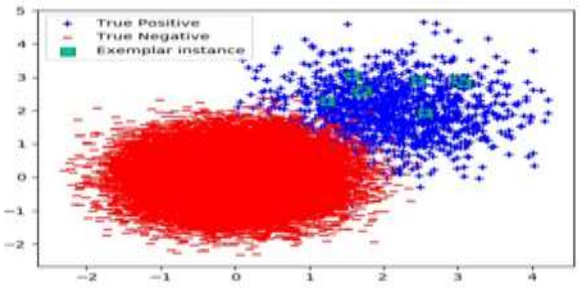

Fig. 5: A set of exemplar for MIL, by OSCE method in100000 data points.

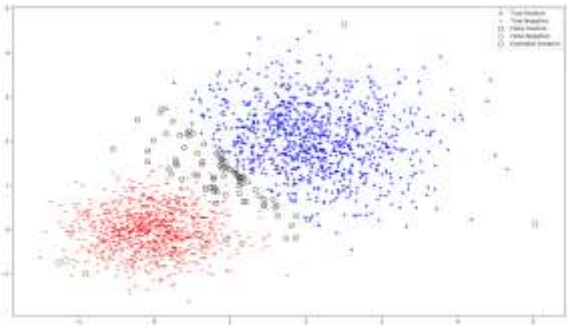

Fig. 6: the true-positive, true-negative, false-positive, false- negative, and a set of examplar in a "2000 data points" withan accuracy of 97 per cent after applying a threshold to find binary η .

4.1. OSCE Classification Results

Fig. 7 indicates the effect of classification with a different number of exemplars in ILP optimization. It concluded that a minimum number of exemplars (here 2 exemplars) is enough to classify the data set. If the radius of a neighborhood is smaller, then the number of exemplars should be more. The recommended size of neighborhood is proportional to the variance of the data points. In some cases, around a query point with a specific radius, there is no neighbor. To have a general solution to calculate similarity metric (w), k=5 nearest

neighbors is suggested instead of the neighborhood radius.

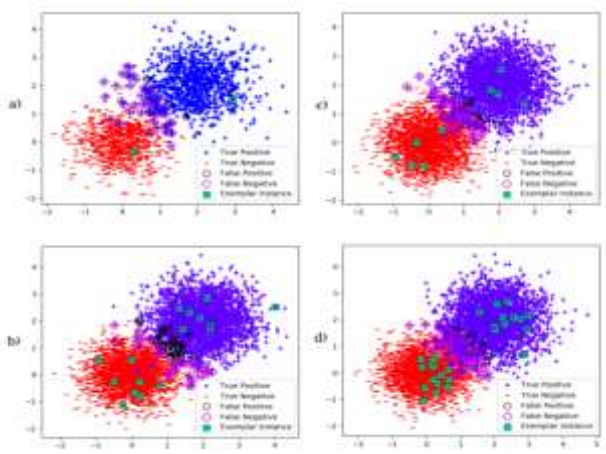

Fig. 7: a) Two exemplar, b) 10 exemplars, c) 20 exemplars, d) 30 exemplars, for ILP in a 2000 data points.

Fig.8 shows the 150 exemplars in LP optimization with M=1 and grater than two ($\sum_{i\in\Omega} \eta_i \leq M$). The sum of the η for M greater than 2 approximately is 1.6. In ILP, the value of M is equal to the number of exemplars. But in LP, the number of maximum η_i after thresholding, represent the number of exemplars. So, for choosing some exemplars, the k maximum of η will be select. In comparison to ILP with LP, LP is more sensitive than ILP. With a lower number of exemplars by using ILP, good accuracy is achievable.

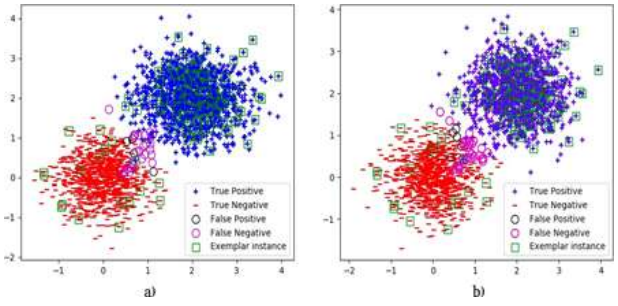

Fig. 8: 150 exemplars for LP in a 2000 data points, a) M=1, b) M>2.

The number of neighbors and exemplars have a significant effect on the accuracy. How many neighbors and exemplars are suitable to find the best accuracy?!

How many neighbors

Different number of neighbors is tested to find the best number of neighbors. The experiments shows that, k>15 is not suitable (Fig. 9).

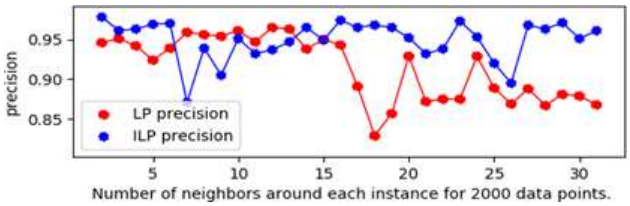

Fig. 9: Precision with different number of neighbors for eachinstance in 2000 data points.

How many exemplars

Comparison of the precision with different number of exemplars illustrates that 2.5% of the number of data points is enough to find optimum result (Fig. 10). In this figure, for more than 50 exemplars, the accuracy reaches the highest point and approximately is fixed for more exemplars.

$$\frac{50 \text{ exemplars}}{2000 \text{ datapoints}} = 0.025$$

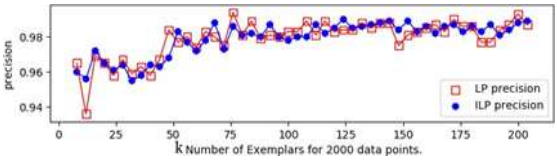

Fig. 10: Precision with different number of exemplars in 2000 data points.

classification run time

In general, either OSCE is faster than the K-NN classifier with k=5. Also, ANN is faster than both of them. It is because of using a set of exemplars instead of hole data points to predict the label. So, in a large data set K-NN is slower. In contrast, the accuracy of

ANN is lower than OSCE and K-NN.

In conclusion, OSCE in a large data set has more ad- vantage. Indeed, there is a trade-off between run time classification and accuracy. Fig. 11 compares the run time classification of OSCE, K-NN and ANN classifier.

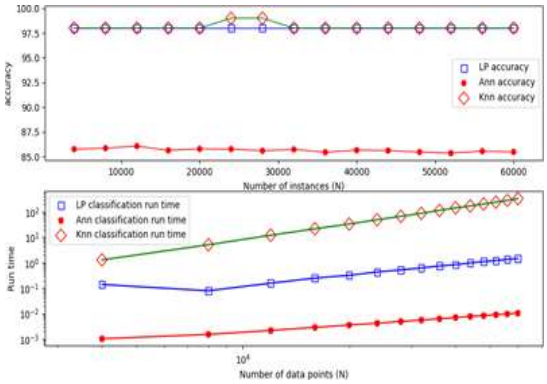

Fig. 11: Run time classification of OSCE, K-NN and ANN classifier.

Figure 11 illustrates the lower recall for K-NN in comparison with the exemplar classifier for larger data sets. In addition, OSCE is faster than neural network (NN) and NN is faster than convolutional neural network (CNN). The testing for the exemplar method is of course faster than trying to find NN in the whole training set. This is obvious and comes from the fact that there are many more data points than exemplars. There is no need to prove it. We are of course paying for the speed by a decrease in accuracy. The question is, how much. So, the only reasonable test is an experiment, where we evaluate the trade-off between accuracy and speed for all competing methods. Another aspect is learning. There, the exemplar method is slower than plain nearest neighbors that have lower accuracy than OSEC (Fig.11).

4.2. MIL by OSCE

The proposed algorithm tested in different data points. If the number of features for each patch increases the threshold for classifying should be changed and increase. Although, the OSCE classifier have been tested in the synthetic data set from the CMP data server in the Czech Republic. The result indicates the success of this algorithm (see Fig.12). In Fig.12 the number of exemplars is 50; And after setting manually the optimal threshold, 100% accuracy achieved. The training run time for the OSCE classifier is very fast. Here we had more than 100000 data points, and the OSCE run time after less than one hour finished.

All in all, in this data set, 100% accuracy is obtained. In general, the exemplar classifier is faster than the K- NN classifier. It is because of using a set of exemplars instead of whole data points to predict the label. So, in large data set K-NN is slower. Figure 11 illustrates the lower recall for K-NN in comparison with the exemplar classifier for larger data sets.

4.3. Discussion

In this reaserch a fast examplar-based learning algorithm needed to be developed, hence, we want a small number of exemplars, as the complexity depends linearly on the number of exemplars. This leads to a constraint optimization problem, where the constraint is the l0 norm, i.e. the number of nonzero weights. Equivalently, it can be also formulated as a binary (or integer) linear program. Usually, in ILP is difficult to optimize, so we relax that problem using linear constraints, obtaining a linear program, which in many cases leads to the same solution. In addition, for implementing the proposed algorithm, Gurobi* (an optimization free library) is used.

Brute force

In order to have a better understanding of the algorithm, it programmed by brute force approach. In a small data set and a few exemplars, there is no need for long run time

* https : //www.gurobi.com/products/gurobi − optimizer/

or memory. But in a larger data set, a different combination of exemplars have been tried randomly. For instance, if 5 exemplars of 1000 data set are suggested, $\frac{1000!}{(1000\text{-}5)!5!} \approx 8\times10^{12}$, a great number of combination is possible. So, like the Gurobi library we used the random generation of k binary number. The result by brute force (Fig. 12) is similar to achieved result by Gurobi.

- Evaluation on digit data set

To test the OSCE classifier in popular and more complex data set, digit data set* is used to evaluate the accuracy of the exemplar classifier (digit one as positive class and other digits as negative class). In conclusion, if set k=10 neighbors for each query points to calculate distance weights (Eq. 3), and the number of exemplars equal to 0.1% of the number of data points, the 99% accuracy is resulted.

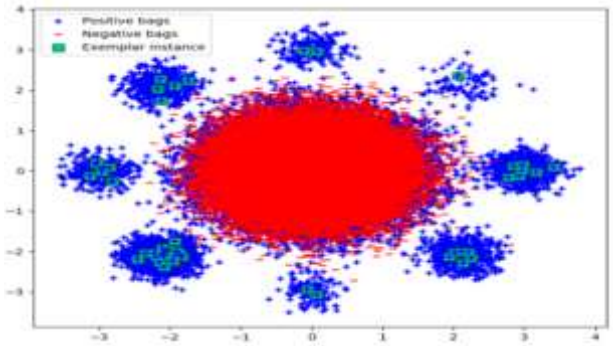

Fig. 12: A set of exemplars for MIL, by OSCE method in 102400 data points.

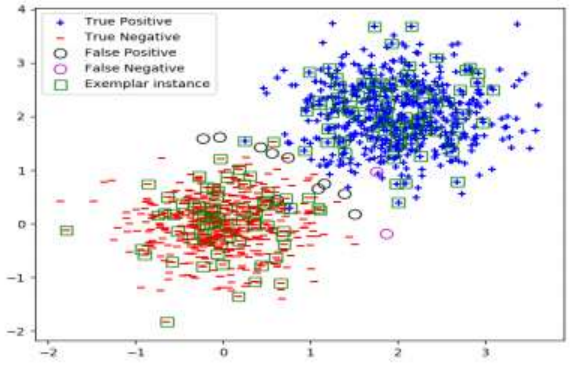

Fig. 13: 150 exemplars in 1000 data points by Brute force.

5. Conclusion

A fast exemplar-based learning algorithm has been developed, for multi-instance learning (MIL) applications. Therefore a small number of exemplars, is been chosen, as the complexity depends linearly on the number of exemplars. This led to a constraint optimization problem where there were four constraints, i.e. the number of nonzero weights. Equivalently, it could be also formulated as a binary (or integer) linear program. Usually, ILP was difficult to optimize, so that problem is been relaxed by using linear constraints, which resulted in obtaining a linear program. Also, this hasled various cases to the same solution results. In a large data set, ILP could not optimize cost function as fast as LP. So, thresholding on the achieved variable by LP is proposed. Also, smaller neighbors radius or smaller k neighbors for each query point yielded to a faster training classifier run time. At last, a comparison with ANN and K-NN cluster had done. Results emphasize that in a large data set, OSCE is better than both clusters. Additionally, it's been discussed that the OSCE was faster than NN and CNN classifiers inlarge data points.

* https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html

Acknowledgments

We thank Mrs Sepideh Zeineddin Pakdaman who is one of the active members of Security Department of Iran, Telecommunication Research Centre (ITRC), Tehran, Iran, for editing and cooperating us in this paper.

References

[1] H. Samet, "Foundations of multidimensional and metric data structures", Morgan Kaufmann, 2006.

[2] R. Weber, H. Schek, and S. Blott,A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces, in Pro- ceedings of the International Conference on Very Large Data Bases, IEEE, 1998, pp. 194–205.

[3] S. Dasgupta and Y. Freund, Random projection trees and low dimensional manifolds, in Proceedings of the 40th annual ACM symposium on Theory of computing, ACM, 2008, pp. 537–546.

[4] P. Jones, A. Osipov, and V. Rokhlin, Randomized approximate nearest neighbors algorithm, Proceedings of the National Academy of Sciences, 108 (2011),pp. 15679–15686.

[5] D. Aiger, E. Kokiopoulou, and E. Rivlin, Random grids: Fast approximate nearest neighborsand range searching for image search, in Computer Vision (ICCV), 2013 IEEE International Conferenceon, IEEE, 2013, pp. 3471–3478.

[6] Y. Li and X. Zhang, "Improving k nearest neighbor with exemplar generalization for imbalanced classification," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6635 LNAI, no. PART 2, pp. 321–332, 2011.

[7] J. Rousu, C. Saunders, S. Szedmak, and J. Shawe-Taylor, "Kernel-based learning of hierarchical multilabel classification models," Journal of Machine Learning Research, vol. 7, no. Jul, pp. 1601–1626, 2006.

[8] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," IEEE Transactions on Geoscience and Remote Sensing, vol. 43, no. 6, pp. 1351–1362, 2005.

[9] D. L. Hintzman and G. Ludlam, "Differential forgetting of prototypes and old instances: Simulation by an exemplar-based classification model," Memory and Cognition, vol. 8, no. 4, pp. 378–382, 1980.

[10] M. P. McANDREWS and M. Moscovitch, "Rule-based and exemplar-based classification in artificial grammar learning," Memory and Cognition, vol. 13, no. 5, pp. 469–475, 1985.

[11] Zhou, Zhi-Hua, and Min-Ling Zhang. "Neural networks for multi-instance learning." In Proceedings of the International Conference on Intelligent Information Technology, Beijing, China, pp. 455-459. 2002.

[12] Zhang, Min-Ling, and Zhi-Hua Zhou. "Improve multi-instance neural networks through feature selection." Neural processing letters 19, no. 1, pp. 1-10, 2004.

[13] M.Liu, J.Zhang, E.Adeli, and D.Shen, "Landmark-based deep multi-instance learning for brain disease diagnosis", Medical image analysis, 43, pp.157-168, 2018.

[14] T.Khatibi, A.Shahsavari, A.Farahani, "Proposing a novel multi-instance learning model for tuberculosis recognition from chest X-ray images based on CNNs, complex networks and stacked ensemble".Physical and Engineering Sciences in Medicine, vol.44, no.1, pp.291-311, 2021.

[15] X.Wang, F.Tang, L.Luo, Z.Tang, A.Ran, C.Cheung, P.Heng "uncertainty-driven deep multiple instance learning for OCT image classification".IEEE journal of biomedical and health informatics, vol.24, no.12, pp.3431-3442, 2020.