Multiple Instance Learning using EMDD

Vivin Paliath, Tejas Khairnar

Arizona State University

IRA A. FULTON SCHOOLS OF engineering
ARIZONA STATE UNIVERSITY

Introduction

- Investigate and implement the **Expectation Maximization Diverse Density** (EM-DD) algorithm
- EMDD is an approach that has seen significant success in Multiple-Instances Learning problems and is a generalization of the supervised-learning classification-problem.
- We demonstrate an understanding of this algorithm by providing a valid implementation and measuring its performance against various data sets.
- Compare the performance of the EM-DD algorithm against the MatLab MIL-toolkit implementation.

Our Approach

- In Python, we used the SciPy Library to perform Mathematical operations
- To minimize the criterion function we used scipy.optimize.minize
- EMDD algorithm in MILL toolkit uses the average of the estimated target and scale vector corresponding to the best diverse density
- We were able to increase our performance by using average of the results.
- Update the value of the previous density only if the current density is better
- Calculate Precision and Recall in addition to the accuracy.
- Perform 10-fold cross validation to calculate average accuracy.

EMDD algorithm

- Step 1: The algorithm starts with an initial guess of the target point t, using instances from positive bags
- Step 2: It then repeatedly performs the E -step and the M -step to search for the maximum-likelihood hypothesis
- Step 3: In the E -step, the algorithm needs to select the appropriate instances from the bag.
- Step 4: It identifies the most-likely instance from each bag that is responsible for the label of the bag, based on the current target t .
- Step 5: In the M -step, the algorithm uses a gradient-descent search to find the new target t' that maximizes the diverse-density at h.
- Step 6: After the maximization set, t is set to t0 and the algorithm returns to the first step and runs until convergence.

Results

TABLE I
COMPARISON OF ACCURACY ON MUSK1 DATA-SET BETWEEN EM-DD IMPLEMENTATIONS AND OTHER MIL ALGORITHMS

Algorithm	Accuracy
EM-DD (paper implementation)	96.8%
EM-DD (our implementation)	86%
EM-DD (MILL implementation)	84.9%
Iterated-discrim APR	91.3%
SVM	80.5%
Citation-kNN	90.4%

TABLE II
COMPARISON OF ACCURACY BETWEEN PYTHON AND MILL IMPLEMENTATION OF EM-DD ON MULTIPLE DATA-SETS

Data Set	Accuracy (Python)	Accuracy (MILL)
MUSK1	86%	84.9%
Synthetic data-set 1	79%	73.25%
Synthetic data-set 4	76%	82.4%
Diabetic-retinopathy data-set	83%	55.05%

Results - con't

TABLE III
ACCURACY FOR PYTHON EM-DD IMPLEMENTATION ON ALL DATA-SETS

Data Set	Accuracy
MUSK1	86%
MUSK2	80.99%
Synthetic data-set 1	79%
Synthetic data-set 4	76%
Diabetic-retinopathy data	83%
Elephant	92%
Tiger	85%
Fox	78%

TABLE IV
PRECISION FOR PYTHON EM-DD IMPLEMENTATION ON ALL DATA-SETS

Data Set	Precision
MUSK1	85.33%
MUSK2	89.66%
Synthetic data-set 1	85.5%
Synthetic data-set 4	80.5%
Diabetic-retinopathy data	86.73%
Elephant	94.4%
Tiger	89.33%
Fox	79.4%

TABLE V
RECALL FOR PYTHON EM-DD IMPLEMENTATION ON ALL DATA-SETS

Data Set	Recall
MUSK1	72.54%
MUSK2	57%
Synthetic data-set 1	70.73%
Synthetic data-set 4	74.75%
Diabetic-retinopathy data	85.9%
Elephant	90.25%
Tiger	79.16%
Fox	73.8%

Conclusions

- Successfully implemented EMDD algorithm in Python
- Ran our implementation over the following datasets
  - Musk 1
  - Musk 2
  - Synthetic Dataset 1
  - Synthetic Dataset 4
  - Diabetic-retinopathy Dataset
  - Elephant dataset
  - Tiger Dataset
  - Fox Dataset
- Our average acuracy across 10-fold cross validation is better than the MIL implementation
- Investigated other methods like Iterated-discrim APR, SVM, Citation-kNN which solve the MIL problem
- In order to further improve the accuracy we should perform more experiments, use a better minimizer, and examine the role of threshold

References

1. Q. Zhang et.al "Em-dd: An improved multiple-instance learning technique," in Advances in neural information processing systems, 2001, pp. 1073–1080.
2. T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez, "Solving the multiple instance problem with axis-parallel rectangles," Artificial intelligence, vol. 89, no. 1, pp. 31–71, 1997.
3. J. Wang and J.-D. Zucker, "Solving multiple instance problem: A lazy learning approach," 2000.
4. Ragav Venkatesan, "Synthetic Dataset for MIL," https://github.com/ragavvenkatesan/np-mil/tree/master/data.