Model fusion-based batch learning with application to oil spills detection

Yang, Chunsheng; Yang, Yubin; Liu, Jie

Publisher's version / Version de l'éditeur:
https://doi.org/10.1007/978-3-642-31087-4_5
Advanced Research in Applied Artificial Intelligence, Lecture Notes in Computer Science; no. 7345, pp. 40-47, 2012-08-01

Model Fusion-based Batch Learning with Application to Oil Spills Detection

Chunsheng Yang[1], and Yubin Yang[2]
[1]Institute for Information Technology, National Research Council Canada
Chunsheng.Yang@nrc.gc.ca
[2] State Key Laboratory for Novel Software Technology, Nanjing University
yangyubin@nju.edu.cn

Abstract: Data split into batches is very common in real-world applications. In speech recognition and handwriting identification, the batches are different people. In areas like oil spill detection and train wheel failure prediction, the batches are the particular circumstances when the readings were recorded. The recent research has proved that it is important to respect the batch structure when learning models for batched data. We believe that such a batch structure is also an opportunity that can be exploited in the learning process. In this short paper, we investigated the novel method for dealing with the batched data. We applied the developed batch learning techniques to detect oil spills using radar images collected from satellite stations. This paper reports some progress on the proposed batch learning method and the preliminary results obtained from oil spills detection.
Keywords: Batch Data, Batch Learning, Transfer Learning, Content-based Learning, Model Fusion, Oil Spill Detection.

1. Introduction

In real-world applications, many data are split into batches, and each batch may have its feature space and distribution. For instance, in speech recognition and handwriting identification, the batches are different people. In the area of oil spill detection [1] the batches are the radar images collected from different satellite stations. In the area like train wheel failure predictions [2], the batches are the particular circumstance from different trains. We believe that it is necessary and important to respect batched structure and its characteristics when learning and evaluating algorithms for batched data in real-world applications. Unfortunately, most existing machine learning or data mining techniques/algorithms assume that the data used in training and in the testing has the same feature space and the same distribution. Such an assumption in many real-world applications hardly holds. Recently, many efforts have been put to transfer learning [7]. Basically, transfer learning resolves the issue that training date and future deployment environment (or so-called testing data) may have different distributions or feature space. Transfer learning does not deal with issues in batched data directly. In this work, we attempt to address this issue and investigate an effective method for batch learning by respecting batched data structure and its characteristics represented by features instead of attributes. We are motivated by Kubat et al.'s work on oil spills detection using radar imaged collected from satellite stations. We proposed a model fusion-based method for batch learning and applied it to oil spills detection. In this short paper, we report the on-going progress along with preliminary results and mainly discus the future directions on batch learning.

The rest of this paper is organized as follows. Section 2 presents a problem formulation on batch learning by reviewing related work such as transfer learning; Section 3 introduces the proposed method; Section 4 outlines the experiments conducted in oil spill detection along with some preliminary results; Section 5 discusses the limitation and provides the future direction; the final section concludes the paper.

2. Batch Learning and Related Work

Over the past decades, many researchers have paid attention to transfer learning [7] which is closely related to batch learning. Transfer learning also has different names in machine learning community, including learning to learning, meta-learning, multitask learning [8], and context-sensitive learning [9], robust learning etc. All of these work tried to address the learning problems from one domain to another domain. In other words, these techniques learn a model in a source domain and apply it to a target domain which may have different feature space or distinct distribution. The ultimate goal is to maximize generalization ability of the learn models. In term of the definition given in [7], transfer learning is defined as follows. Given two different domains: source domain (D_s) and target domain (D_T) with learning task T_s and T_T, respectively , transfer learning aims at improving the ability of learning algorithms in D_T using the knowledge in D_T and T_s where $D_s \neq D_T$ and $T_s \neq T_T$. In terms of the definition, there are three different settings which lead to three main transfer learning techniques: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning. Inductive transfer learning deals with the learning issues where the learning tasks are different no matter what domains are the same or not; transductive transfer learning focuses on the problems where domains are different but the target tasks are the same; and unsupervised transfer learning are similar with inductive learning, but the learning tasks in target domain is related to ones in the source domain.

However, batch learning is different from transfer learning. Batch learning is defined as follows. Given that a domain is decomposed into multiple sub-domains expressed with batched data, batched learning is to find the best learning algorithm for domain problem. If we note the batched data as $D_1, D_2, D_i \ldots D_n$ (n is the batch number) and each batch data has its own feature space(note as F_i) and probability distribution (fd_i), batch learning aims to find the best learning algorithm (noted as $f(\cdot)$), $f(\cdot) = \mathrm{F}(F_1, F_2, F_i \ldots F_n)$, where $F_i \neq F_j$ or $fd_i \neq fd_j$. In batch learning, the learning task is always the same. For example, in oil spills detection, the learning task is always to classify the images into positives or negatives. To respect the batched data structure in developing learning algorithms or techniques, most research work published focused on feature transformation or weighted features in learning process and to build a global classifier to resolve the batched problem. For example, Vural et al. [6] developed a batch classifier with different algorithms: probabilistic analysis and mathematical programming, and applied the developed batch classifier to medical domain: to classify the medical images such as CT scan and MRI image. Similarly, some other research also focused on developing a global model for batch learning [4, 5]. In this work we propose a model fusion-based method by respecting the nature of batched data and

developing model for each individual batch. In the following section, we detail the model fusion-based method for batch learning.

3. Model Fusion-based Framework

We assume that the features extracted from each batch may be different or distribution may be different even through the attribute space in each batch may be identical. We also assume that the batched data covers sufficient samples for the problem space given a domain. Building on techniques from ensemble classifier or model fusion, we developed a model fusion-based method for batch learning. The idea is to build a model (noted as m_i) or classifier for each batch by using the extracted features and then build a batch identifier (noted as m^b) for batch identification using all dataset. The developed method is described in Table 1 using the notation defined above. The method consists of three main steps: feature extraction($F^x(Di)$), batch model building (*BuildModel()*), and batch identifier building (*BuildBatchIdentifier()*).

Table 1, The Model fusion-based method for batch learning

Input: *A batched dataset* $DS = \{D_1, D_2, D_3 ... D_i ... D_n\}$

Output : *models and batch identifier (* m_i *and* m^b *)*

```
Process: {
  For each Di in DS {
      Fi = F^x(Di)                                   /* extracting features for each batch */
          For adopted algorithm j {
              mij = BuildModel( Fi, algorithm j )    /* building model for each batch */
          }
      mi= ModelSelection(mi1, mi2,... mij);
    }
    For specified classifier algorithm i {
      LabelAllbatcheddata(DS)              /* label batched data as a N-class classification task */
      m^b = BuildBatchIdentifier(algorithm i, DS )   /* building batch identifier*/
    }
   OutputModels(mi, m^b )
}
```

First step is to extract the features which represent the batch characteristics for each batch data. We developed the automated approach that could help further improve the data representation in a batch by creating new powerful features that combine the initial ones. For each batch the created new features may be different, which make it possible to respect the batched data structure in batch learning.

The second step is to independently build model for each batch using the extracted features. To do so, many learning algorithms are available including Instance-Based learning (IB*k*), TFIDF classifier, Naïve Bayes, Support Vector Machine (SVM), Decision Trees, and Neural Networks. In this work, we tend to prefer simple algorithms such as Decision Trees and Naïve Bayes over more complex ones because they are quick and produce models that we can easily explain for each batch. For the built models we evaluate their performance with traditional matrix such as accuracy and select the best one for each batch.

Third step is to build a high-performance classifier as batch identifier. This is N-class classifier. We first label all data in the batched dataset. In other words, we label the data in batch D_i as the class '***i***' , and we will have N-class dataset finally. Using this dataset, we can build a N-class classifier as the batch identifier. We just need to repeat the model building process in the second step and find one best classifier for given batched data.

After developing the model for each batch and building a batch identifier, we can build a model fusion-based classification system for real-world applications. The Figure 1 shows such a classification system for batched applications. Given an observation ($\vec{X}$), it will be identified first as which batch it belongs to. Then corresponding batch model will be selected to classify the given observation and final result will be generated from the batched classification systems.

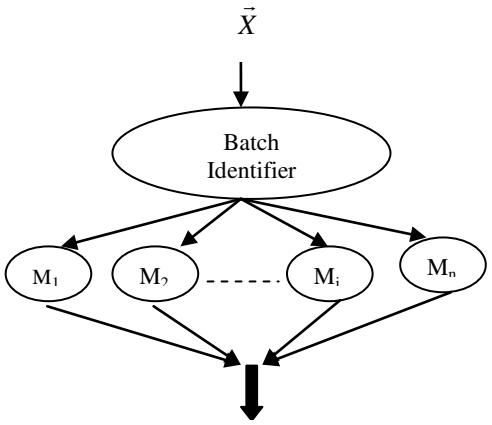

Figure 1, the structure of Batched Classification

4. Experiments and Results

In this section, we demonstrate how the developed batch learning technique works by applying it to oil spills detection. We start from the brief description of the oil spills detection application, then present the experiment conducted along with some preliminary results.

4.1 The Application Domain

Oil spill is an important environment problem. It is interesting the majority of oil spills is caused by ships which depose of oil residues in their tanks during navigation [1]. Radar images collected from satellites stations provide a vital resource to detect oil spills. Such a radar image contains 8,000X8,000 pixels with each pixel representing a square of 30X30m. Oil slicks will appear dark in the image and its size and shape change in time depending on weather and sea conditions. In terms of the nature of image data, such image data is

batched. Those images can be used to detect oil spill. Kubat et al. [1] applied machine learning techniques to detect oil spills, pushing oil spills detection toward advanced step from manual detection. In the early satellite image service such as the Tromso Satellite Station (TSS) service in Norway, the oil spills are distinguished by well-trained human expert who can distinct the images between non-spills and spills. Kubat et al. built one global model (classifier) for nine batched data. Even though the developed classifier could help to detect oil spills with much better accuracy than human does, Kubat et al still pointed out the new methods for batch learning are demanded. In this work, we applied our proposed batch learning technique to build a model fusion-based classification system for oil spill detection.

4.2 Experiments and Results

We have obtained the dataset for oil spill detection. This dataset consists of 9 batches. The Table 2 shows the composition of each batch: the number of positive and negative instances for each batch. Each batch data contains 54 attributes covering three groups: target measurements, relative measurements, and non-imagery information.

Table 2, The numbers of positive and negative instances in the batched image data

Batched Images	1	2	3	4	5	6	7	8	9	All
Positives	8	4	2	6	2	4	3	5	7	41
Negatives	3	180	101	129	60	70	76	80	197	896
Total	11	184	103	135	62	74	79	85	204	937

Table 3, the accuracy of learning algorithms, J48 and Naïve Bayes

Batches	J48 (Decision Tree)		Naïve Bayes	
	Model Fusion	One Model	Model Fusion	One Model
1	0.91	0.27	0.91	0.27
2	0.995	0.17	0.96	0.08
3	1.0	0.09	1.0	0.78
4	0.99	0.14	0.92	0.04
5	1.0	0.95	1.0	0.71
6	0.98	0.89	0.74	0.93
7	0.98	0.95	0.93	0.61
8	0.99	0.98	0.93	0.87
9	0.99	0.90	0.96	0.95
Total	0.99	0.77	0.94	0.62

We first extracted the correlated features from each batch for building models for individual batch. In this work, we employed decision tree (J48) and Naïve Bayes as the learning algorithms for developing model for each batch.

In order to evaluate the effectiveness of the model fusion-based approach for batch learning, we conducted two different experiments: developing model fusion-based classification systems and building one global model for all batches. To compare the performance of two different systems, we performed evaluation using cross valuation method. Because of small size of the data, we only used 5 folds cross validation. Table 2 shows the results. We calculated the accuracy for each batch and a total for all batches. From the Table2, it is obvious that model fusion-based approach outperforms one global model approach for both selected learning algorithms.

5. Discussions and Future Work

The results from oil spills detection demonstrated the effectiveness of the proposed model fusion-based method for batch learning. By respecting the batched data structure in the real-world application, batch learning is a useful technique for batched data. Even though we emphases the importance of batch learning though oil spills detection application, the issues facing us still is a generic one in machine learning. This paper just reported some progress. Many tasks are on-going and will continuously be our future work as well. These issues are as follows.

(1) When the number of batches becomes large the proposed method will be more difficult to deal with the batched issue. In the proposed method, the performance of batch learning systems depends largely on the performance of a batch identifier. The larger the number of batches, the bigger the number of classes for N-class classifier. This will increase the difficulty to build a high-performance classifier. In oil spill detection, we developed a 9-class classifier to identify an observation into one of nine batches. The results shown in Table 2 rely on the 98% accuracy of 9-class batch identifier in this application. However, it is not easy to build such a high-performance classifier with the increasing of the number of batches. To address this issue, we can combine the transfer learning technique and model fusion-based method. We can reduce the batch numbers by combining or grouping the number of batches into a new batch with the help of transfer learning technique. In other words, after examining the feature space and their distribution of the batches, we can transform the feature space or distribution to get a unified feature space or identical distribution for some batches, such that the number of batches will be reduced.
(2) In this work, we developed the model for each batch using the same learning algorithm either J48 or Naïve Bayes. In fact, we can evaluate different learning algorithms for each individual batch to find out the best matching model. This will be the next experiments to look at the model diversity issue. To this end, we need to use more sophisticated model evaluation method such as ROC curve and cost-based curve and integrate the batched feature into evaluation as well.

(3) When we investigated batch leaning techniques for batched data, we assume that the number of batches (the number of sub-problems) is known from real-world applications. For example, the oil spills detection contains nine batches. This assumption may hardly hold because new batch may appear anytime. In such a case, the proposed method has difficult to deal with new batch. We have to rebuild model for new batch and rebuild batch identifier as well. To address this issue, robust learning technique is demanded. Therefore, we are looking into the method for integrating robust leaning with batch learning.

6. Conclusions

In this paper, we emphasized that it is necessary to deal with batched data by respecting the batched data structure. Building the techniques of classifier ensemble and machine learning, we developed a model fusion-based method for batch learning. We also applied the proposed method to a real-world application: to detect oil spills using the batched radar images data. The preliminary results demonstrated the feasibility and usefulness of the proposed method for batch learning. As we mentioned, we only reported some progress for our on-going research on batch learning. We also discussed several critical issues in dealing with batch learning and these issues will be our future work.

Acknowledgment

The authors would like to thank Dr. Robert C. Holte for providing data from the oil spills detection and allow us to explore it more deeply. We also like to thank Dr. Chris Drummond for his valuable discussion and comments on experiments and future direction. This work is supported by the Natural Science Foundation of China (Grant Nos. 61035003, 61021062, 60875011), the International Science and Technology Cooperation Program of China (Grant No. 2010DFA11030), and the Natural Science Foundation of Jiangsu, China (Grant No. BK2011005, BK2010054).

Reference

[1] M. Kubat, R. Holte, and S. Matwin, “Machine Learning for the Detection of Oil Spills in Satellite Radar Images”, Machine Learning, 30, 195-215, 1998
[2] C. Yang and S. Létourneau, "Learning to predict train wheel failures", The 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 516-525, 2005
[3] S. Bermejo and J. Cabestany, “A Batch Learning Vector Quantization Algorithm for Nearest Neighbor Classification”, Neural Processing Letters, 1!:173-184, 2000
[4] M. Kinouchi, N. Takada, Y. Kudo, and T. Ikemura, “Quick Learning for Batch-learning Self-Organizing Map”, Genome Informatics 13:266-267, 2002
[5] K. Y. M. Wong, P. Lou, and F. Li, “Dynamics of Gradient-based Learning and Application to Hyperparameter Estimation”, IDEAL 2003, LNCS 2690, pp. 369-376
[6] V. Vural, G. Fung, B. Krishnapuram, J. Dy, and B. Rao, “Batch Classification with Application in Computer Aided Diagnosis”, ECML 2006, pp. 449-460, 2006
[7] S. J. Pan and Q. Yang, “A Survey on Transfer Learning”, IEEE Transaction on Knowledge and Data Engineering, Vol. 22 No. 10, pp. 1345-1359, 2010
[8] R. Caruana, “Multitask Learning”, Machine Learning, Vol28, pp.41-75, 1997
W. W. Cohen and Y. Singer, “Context-Sensitive Learning Methods for Text Categorization”, ACM Transactions on Information Systems, Vol. 17, No. 2, pp. 141-173, 1999