REPORT DOCUMENTATION PAGE

Form Approved

OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
09-03-2015	Final	27-03-2013 – 26-03-2015

4. TITLE AND SUBTITLE

A Large-scale Distributed Indexed Learning Framework for Data that Cannot Fit into Memory

5a. CONTRACT NUMBER
FA2386-13-1-4045

5b. GRANT NUMBER
Gtant AOARD-134045

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Shou-De Lin

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Computer Science & Information Engineering Department
National Taiwan University
No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AOARD
UNIT 45002
APO AP 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AOARD-134045

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Code A: Approved for public release, distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This project deals with issues on distributed learning for big data and addresses three major problems. 1) Learning a classifier where data contain many samples that do not help improve the model quality, which cost much I/O and large memory to process. A Block Coordinate Descent combined with Approximate Nearest Neighbor (ANN) search to select active samples in dual mode was shown to outperform the-state-of-the-art. 2) Complex query search in which sending it to all the local machines is very costly. Decomposing the reference patterns into multi-resolution solved the distributed kNN/kFN pattern matching very efficiently. 3) Distributed learning problem for unlimited unlabeled data stream from many clients needed to send to a server to learn a classifier. Integrating three learning techniques (online, semi-supervised and active learning) together with a selective sampling with minimum communication between the server and the clients solved this problem.

15. SUBJECT TERMS

Distributed computing, Client, Server, Classification, Nearest neighbor, Block coordinate descent, Active sampling, Semi-supervised learning, Communication

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Hiroshi Motoda, Ph. D.
U	U	U	SAR	10	19b. TELEPHONE NUMBER *(Include area code)* +81-42-511-2011

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
26 MAR 2015	**Final**	**27-03-2013 to 26-03-2015**

4. TITLE AND SUBTITLE
A Large-scale Distributed Indexed Learning Framework for Data that

5a. CONTRACT NUMBER
FA2386-13-1-4045

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Shou-De Lin

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Science & Information Engineering Department,National Taiwan University,No. 1, Sec. 4, Roosevelt Rd.,,Taipei 10617, Taiwan,NA,NA

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AOARD, UNIT 45002, APO, AP, 96338-5002

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR/IOA(AOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AOARD-134045

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This project deals with issues on distributed learning for big data and addresses three major problems. 1) Learning a classifier where data contain many samples that do not help improve the model quality, which cost much I/O and large memory to process. A Block Coordinate Descent combined with Approximate Nearest Neighbor (ANN) search to select active samples in dual mode was shown to outperform the-state-of-the-art. 2) Complex query search in which sending it to all the local machines is very costly. Decomposing the reference patterns into multi-resolution solved the distributed kNN/kFN pattern matching very efficiently. 3) Distributed learning problem for unlimited unlabeled data stream from many clients needed to send to a server to learn a classifier. Integrating three learning techniques (online, semi-supervised and active learning) together with a selective sampling with minimum communication between the server and the clients solved this problem.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**10**	

Final Report for AOARD Grant FA2836-13-1-4045

"**Research Title**" A Large-scale Distributed Indexed Learning Framework for Data that Cannot Fit into Memory

Date 3/27/2015

Name of Principal Investigators (PI and Co-PIs): Shou-De Lin
- e-mail address : sdlin@csie.ntu.edu.tw
- Institution : National Taiwan University
- Mailing Address : No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan
- Phone : +886-33664888-333
- Fax : +886-2-33664898

Period of Performance: 3/27/2013 – 3/26/2015

Abstract: This work aims at dealing with issues on distributed learning given big data. It contains three major works.

(1) We deal with the situation when data contain large amount of samples that do not help improve the quality of model, but still cost much I/O and memory to process (published in KDD2013). In this research, we show how a Block Coordinate Descent method based on Nearest-Neighbor Index can significantly reduce such cost when learning a dual-sparse model. In particular, we employ truncated loss function to induce a series of convex programs with superior dual sparsity, and solve each dual using Indexed Block Coordinate Descent, which makes use of Approximate Nearest Neighbor (ANN) search to select active dual variables without I/O cost on irrelevant samples. We prove that, despite the bias and weak guarantee from ANN query, the proposed algorithm has global convergence to the solution designed on entire dataset, with sublinear complexity each iteration.

(2) We propose a framework to solve distributed kNN/kFN pattern matching (published in ICDM2013). In the scenario when the query is more complex, the communication cost for sending it to all the local machines for processing can be very high. Our research aims to address this issue by decomposing the reference patterns into a multi-resolution representation. Using novel distance bound designs, our method guarantees the exact results in a communication-efficient manner.

(3) We consider a novel distributed learning problem: A server receives potentially unlimited data from clients in a sequential manner, but only a small initial fraction of these data are labeled. Because communication bandwidth is expensive, each client is limited to sending the server only a small (high-priority) fraction of the unlabeled data it generates, and the server is limited in the amount of prioritization hints it sends back to the client. We present a novel framework for solving this learning problem in an effective and communication-efficient manner. On the server side, our solution combines two diverse learners working collaboratively, yet in distinct roles, on the partially labeled data stream. A compact, online graph-based semi-supervised learner is used to predict labels for the unlabeled data arriving from the clients. Samples from this model are used as ongoing training for a linear classifier. On the client side, our solution prioritizes data based on an active-learning metric that favors instances that are close to the classifier's decision hyperplane and

yet far from each other. To reduce communication, the server sends the classifier's weight-vector to the client only periodically.

Introduction:

We will provide the introduction for the three works we have accomplished separately.

Linear classification has become one of the standard approaches dealing with large-scale analysis in pattern recognition and data mining. Recent advances in training linear model have achieved complexity linear to the data size, where problem with several gigabytes of data can be solved in reasonable time. With the development of more and more efficient algorithms, the learning bottle-neck has shifted from computation to the I/O between disk and memory. The situation becomes especially critical when data cannot put into memory, where repeated data access through comparatively expensive I/O could encumber the performance of any efficient algorithm. How to reduce such I/O cost becomes a focus in recent research on Aim of the research. This research aims to demonstrate how a Nearest-Neighbor index can improve the I/O efficiency in large-scale learning, especially when memory is limited. In practice, this is beneficial since many state-of-the-art indexing methods for Approximate Nearest Neighbor (ANN) search (e.g. Locality-Sensitive Hashing, Metric Tree etc.) do not require repeated data access, and thus only need small memory and one pass of data loading to be built. Furthermore, an index can often be reused for models trained on the same data. Scenarios such as parameter tuning, cross-validation, multi-class classification, feature selection, and data incremental learning all require executing the training algorithm multiple times.

Furthermore, pattern matching in distributed environments is generally considered as a challenging but important task for applications relevant to machine-to-machine (M2M) systems. In such settings where a large amount of local machines are involved in computation and storage, a primary goal is often to minimize the amount of communication needed to compute the answer. Therefore, our research aims at advancing the current state-of-the-art on distributed pattern matching from 'single reference pattern' to 'multiple reference patterns', and proposes a general framework to handle both k nearest and farthest neighbor search of the multiple reference pattern set, while significantly reducing the communication cost, mainly the bandwidth consumption. We present MsWave, a general communication-efficient framework to identify both kNN and kFN instances given multiple time series reference patterns in a distributed environment. To our knowledge, this is the first solution proposed for such purpose. We propose to use average, closest, and furthest neighbor distance to process multiple query (dis)similarity. We then take advantage of the multiple-resolution property of wavelet coefficients, and then for each distance measurement we derive upper and lower bounds of the similarity between each candidate time series to the query set. Such bounds can be exploited to prune candidates for more efficient search without compromising correctness. Moreover, in further contrast to prior approaches, we propose to shift the bounds computation from the server to the local machines to further reduce the bandwidth consumption.

Finally, This paper considers a setting where a set of distributed clients each generate an ongoing stream of data and a server seeks to learn a model of the data. We impose two practical limitations on the setting. First, because of the costs of having humans label large quantities of data, we assume that only a small fraction of the data are labeled. In particular, we focus on a setting where only the first, e.g., 2% of the training data are labeled. Second, because communication bandwidth is often expensive and battery-draining (e.g., a mobile device on a cellular network), we seek

communication-efficient solutions such that each client is limited to sending to the server only a small fraction of the unlabeled data it generates, and limited in how much information it receives from the server. An elegant solution to these problems will face many challenges. First, the amount of data generated by clients can be huge, and even potentially unlimited. As a result, the vast majority of data on the server are unlabeled. Typically, it is not sufficient to train a model with a good generalization ability based merely on limited labeled data. Second, when the volume and velocity of data is high, it is very costly and impossible to store all data either on clients or the server. Thus, traditional approaches that first store data and then train on a static collection are not appropriate in this case. Third, transmitting massive data on the network is discouraged in practice, especially when the network bandwidth is restricted or the communication cost is expensive (e.g., on a cellular network). At first sight, this learning problem seems to share some characteristics of online, semi-supervised, and active learning, which have been extensively studied in the machine learning community. However, it should be noted that our setting differs from these traditional learning settings and may require evolutionary changes to existing algorithms. Unlike online learning problems where all training data are assumed to be labeled, there is only a limited amount of labeled data in our setting. It also differs from typical semi-supervised learning where all labeled and unlabeled data is available ahead of time. Moreover, it differs from standard active learning in that there is no oracle available for providing feedback. Although both settings involve selective sampling, their intentions are different: active learning aims to save labeling efforts, whereas we attempt to reduce the bandwidth consumption between the server and clients (while also keeping the labeling effort to only a small fraction of the data). By considering online, semi-supervised, and active learning jointly, our goal is to develop a modular framework for learning from a remote partially labeled data stream while reducing the bandwidth consumption.

Experiment:

- For the first work:

We conduct several experiments that compare our algorithm (Index-L1-Dual, Index-L2-Dual, and Index-L2-Primal) with state-of-the-art linear SVM solvers LIBLINEAR (L1-Dual, L2-Dual, and L2-Primal), online Pegasos (Online-L1 and Online-L2), and truncated-loss batch solver (Trunc-L1-Dual, Trunc-L2-Dual, and Trunc-L2-Primal) that uses the same truncated-loss function as our method, but employs LIBLINEAR as inner procedure for each convex relaxation.

In limited memory condition, we compare Indexed Block Coordinate Descent with online Pegasos and LIBLINEAR-CDBLOCK (Block-L1-Dual, Block-L2-Dual) which is a limited-memory version of LIBLINEAR. The initialization for both truncated-loss solvers uses 10,000 random samples solved by the corresponding convex loss solver in LIBLINEAR. In our experiments, both I/O and Initialization are included into training time.

Our experiments conducted on 4 large-scale public datasets of increasing size: Covtype, Kddcup1999, PAMAP and Mnist8m. Their statistics are summarized in Table 1.

Table 1: Statistics of Data.

DATASET	#SAMPLES	#FEATURES	STORAGE (KB)
COVTYPE	581,012	54	69,516
KDDCUP1999	4,898,431	126	725,180
PAMAP	3,850,505	104	2,198,880
MNIST8M	8,100,000	784	19,042,640

Table 2 shows the statistics of index built. The construction time and storage size are generally linear to the data size.

Table 2: Statistics of Index.

Dataset	Storage (KB)	Tree Size	Tree Width	Build Time (s)
Covtype	446,444	2,000	10	11
Kddcup$_{1999}$	1,476,580	100,000	100	163
PAMAP	4,554,208	100,000	10	301
Mnist8m	20,704,784	10,000	10	1,539

- Experiments for the 2nd work:

Our experiments consider five frameworks: (i) CP, the Con-current Processing baseline; (ii) PRP, the Probabilistic Processing method; (iii) LEEWAVE-M, which is the state-of-the-art method; (iv) MsWave-S (our proposed framework); and (v) MsWave-L (our proposed framework) and compare the total bandwidth cost for these five frame-works in a distributed environment simulated in MATLAB. We also study the influences on the bandwidth cost of the size of the reference set |Q|, the time series length T, the number of machines m, and the k for kNN/kFN.

We use one real data set and one synthetic data set in our experiments. For the real data set, we choose a public dataset recording the daily average temperature of 300 cities around the world acquired from the temperature data archive of the University of Dayton. The data from each city is considered as a time series with 2048 data points. For the synthetic data set, we use a random walk data model. Each time series is generated by the random walk whose every step size is a normal distributed random number with mean=0 and standard deviation=1. There are 12,500 time series of length 12,500 generated.

There are two strategies we employ to choose the time series that comprise the instances in a query set Q. For the analogous reference set, we choose one time series randomly and then choose its closest |Q|-1 neighbors to form Q; thus the queries in Q are highly similar. For the random reference set, we choose |Q| time series at random.

- Experiments for the 3rd work:

Two experiments were conducted. One to evaluate the server's strategy and one to evaluate the clients'. We used seven data sets downloaded from either the UCI ML repository (wearable, skin) or the LIBSVM website (mushroom, mnist, webspam, gisette, ijcnn1). The motion recognition data set wearable and digit recognition data set mnist were converted into a set of binary problems, respectively, where each class is discriminated against every other class. Totally, we produced 10 problems from wearable and 45 from mnist. For each data set, we balanced the number of instances of each class and linearly rescaled the feature values into the range $[-1, 1]$. We evaluated the algorithms using a set of trials with different partitions of the training and test data. In each trial, we randomly held out half of the data for testing; all instances in the test set were labeled by the algorithms. The remaining data was used for training, of which only a small amount was labeled. Both training and test sets were class-balanced. Next, we randomly permuted the training data and kept labeled data always at the beginning. All algorithms were then incrementally trained with the same permutation in each trial. For evaluation, we paused the training at regular intervals, computed the output hypothesis so far, and calculated its test accuracy. The initial 2% of the training instances are labeled. The size of the candidate pool on the client was 50, from which 10 instances were submitted to the server (a 20% sampling rate). In particular, the following methods were compared in this experiment.

none No unlabeled instances are uploaded to the server. The server stops learning right after labeled instances. Assuming that unlabeled instances can provide useful information, then this approach should give the worst performance.

full All uploaded instances are labeled by an oracle. Intuitively, this approach should give the best result due to the availability of full information.

knn The server employs k-nearest neighbors algorithm, where k = 5. The training set is built by first including all labeled instances, and then adding unlabeled instances with its corresponding predicted labels.

scw (confidence-weighted classifier): The server consists of an SCW model only, which "learns" each unlabeled instance using its own prediction.

knn+scw The server consists of a two-learner model: knn followed by scw. The prediction of knn is used for training scw.

hs+scw Our proposed two-learner model on the server.

hs+scw+cut Our proposed hs+scw model with cutoff averaging for predicting test data.

In the 2nd experiment, fixing the model on the server as hs+scw+cut, we compare the following strategies on the client side.

all All unlabeled instances are uploaded without selection. This incurs 5x the communication costs versus other approaches.

rand Randomly selects instances for uploading.

certain The most certain instances according to the current

server model w are uploaded. The score is defined as $|x^T w|$. This method is similar in spirit to [25].

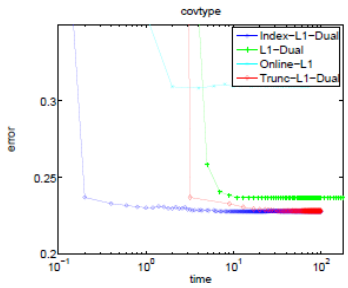
uncertain The most uncertain instances are uploaded.

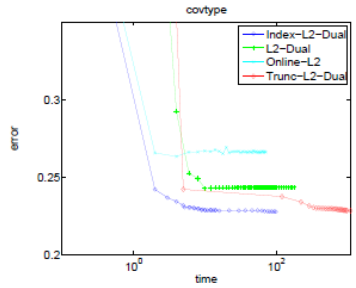
submod Selection is done by optimizing the submodular function that simultaneously considers the uncertainty and redundancy.

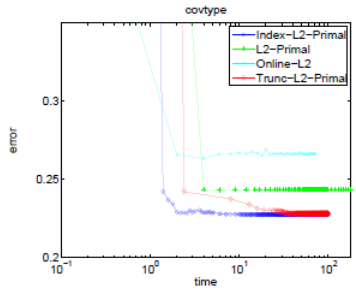
Results and Discussion:

- For the 1st work:

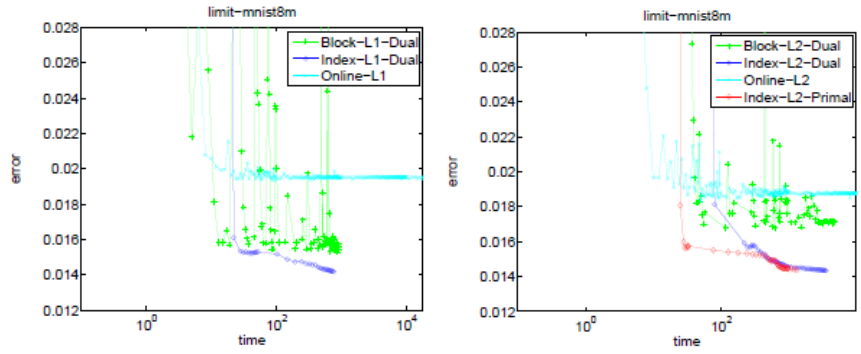
Figures below show the testing error of (L1-Dual, L2-Dual, L2-Primal) solvers under sufficient memory with dataset COVTYPE, where Indexed Block Coordinate Descent saves much I/O time by selecting only relevant samples into memory. Unlike online solver, it converges to much more accurate solution as that produced by truncated-loss batch solver. Though truncated-loss learning problem is non-convex, where different solvers may converge to different solutions, the indexed solver achieves similar accuracy as the truncated-loss batch solver.







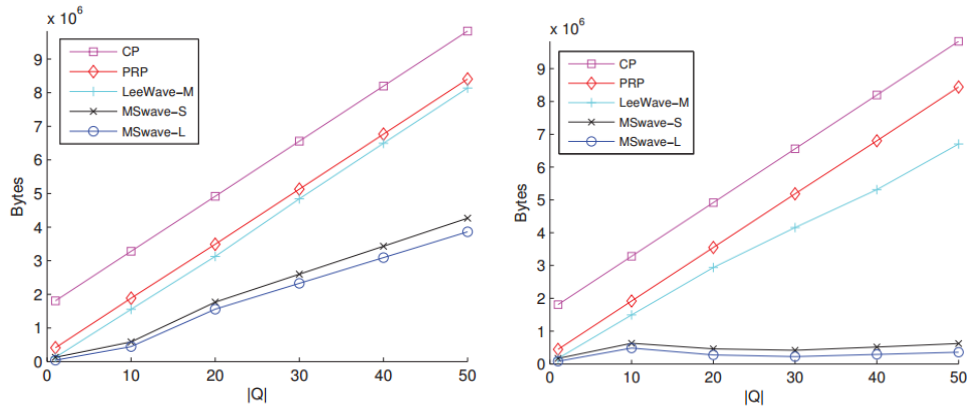
- Figures below shows the limited-memory experiments conducted on Mnist8m. We compare Indexed Block Coordinate Descent with Online Pegasos and LIBLINEAR-CDBLOCK, where data size is 10 times larger than memory space, under which the batch version of LIBLINEAR and truncated-loss solvers suffer from severe swaps and can hardly progress. For Mnist8m, we limit memory size to 2GB, and uses 20 blocks with 1GB cache for LIBLINEAR-CDBLOCK. The Indexed Block Coordinate Descent has almost the same performance as in sufficient-memory condition, where it achieves higher accuracy by selecting informative samples under truncated-loss into memory. Though the size of data was 10 times larger than the memory size, the indexed solver is not affected much since the memory is still large enough for maintaining only relevant samples.



In many applications, large-scale data contain only some relevant samples that can effectively improve the accuracy of model. While random sampling or online learning can overlook those rare but crucial samples, batch learners generally cost too much memory and I/O time. In this research, we propose Indexed Block Coordinate Descent algorithm that makes use of Approximate Nearest Neighbor (ANN) search to select active dual variables without I/O cost on irrelevant samples. Though building index takes time linear to the data size, in practice, this is beneficial since people often learn several models from the same data. Scenarios such as parameter tuning, model selection, cross-validation, multi-class classification, feature selection, and data incremental learning all require multiple passes of training. In the case of limited memory, our approach can save much I/O cost since building index do not require much memory and only requires a pass of data reading. This indexed learning approach can be potentially apply to a general class of large-scale learning problem, through designing new truncated-loss function for convex-loss problems in classification, regression or clustering.

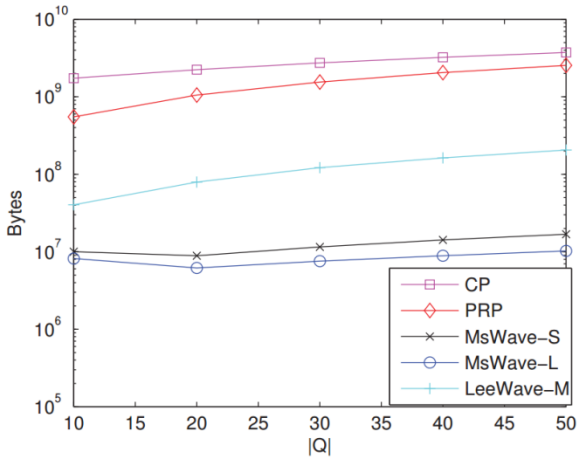
- For the 2nd work

From the figures below, we can see both MsWave-S and MsWave-L outperform the other frameworks significantly for random and analogous reference sets on real data sets. For the analogous reference sets, we also find that the bandwidth costs of the MsWave frameworks do not increase significantly when |Q| increases, regardless of which linkage distance is chosen. Moreover, the performance of MsWave-L is clearly better than MsWave-S and we have proved this difference in bandwidth savings would increases linearly with |Q|.



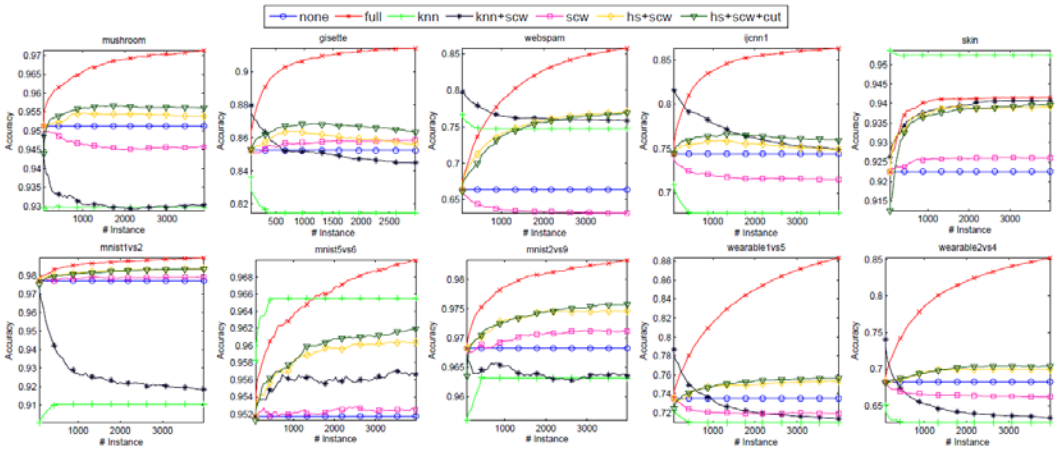
(a) random reference set (b) analogous reference set

The figure below indicates that for large-scale synthetic data, the bandwidth savings for MsWave-L and MsWave-S are even more dramatic, about 1 to 2 orders of magnitude, compared to CP, PRP, and LEEWAVE-M. MsWave' s advantage is fairly consistent across the range of |Q|. In addition, we also observe the gap between MsWave-L and MsWave-S increases as |Q| increases, again agreeing with our analysis.

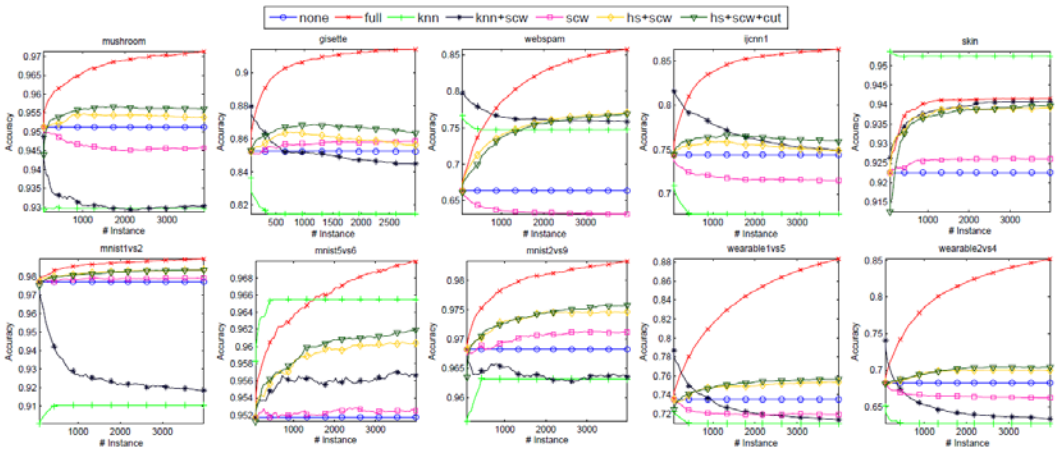


Technically speaking, compared with centralized nearest neighbor search for time series, distributed time-series matching has been studied by only a few prior works, none of which considered more complex query patterns such as multiple time series. Although this paper advances the state-of-the-art by introducing the multiple-series query, we believe there are still many unresolved issues to be explored. For example, we would like to investigate how to improve the response time of such queries, which is constrained by the current one-level-at-a-time approach; how to extend the proposed distributed time series matching mechanism to supervised/semi-supervised learning in a distributed environment; how to extend MsWave to other types of distance measures such as dynamic time warping; and how to resolve other types of complex queries such as "find instances similar to at least k reference instances." These and other open questions make for promising directions for future work.

- For the 3rd work



From above figure it can be observed that proposed hs+scw and hs+scw+cut enjoy superior performance on 8 out of 10 problems comparing to other partial label competitors. On 45 mnist problems, hs+scw and hs+scw+cut yielded on average 0.966 and 0.971 accuracy, respectively. On 20 wearable problems, hs+scw and hs+scw+cut gave 0.699 and 0.714 accuracy, respectively. They are consistently better than the single-learner counterpart scw on all data sets. This indicates the effectiveness of leveraging manifold information of the graph. In fact, on webspam, ijcnn1 and wearable, scw is even worse than none. On webspam, its test accuracy starts with 0.658, decreasing over time and finally yielded 0.637. This is due to the fact that scw completely relies on its own prediction for learning. When the labeling rate is small, the initial hypothesis constructed by labeled data may not be accurate enough. As a consequence, the prediction of scw on the new instance is likely to be wrong, which in turn might mislead the learning procedure. The knn-based approaches, which employ majority voting based on local information, did not show consistent performance. On gisette, webspam, and ijcnn1, the test accuracy of knn decreases until the maximum number of training instances is reached, whereas on mnist it increases. This indicates that a simple bootstrapping for knn is not robust. Also note that, it is not straightforward to formulate a communication-efficient selection policy for knn due to its nonparametric nature. The idea of using the prediction of knn to teach scw is not effective, often resulting in degraded performance of scw over time.



It is interesting to see that all, which transmits all unlabeled data, does not lead to better performance. In fact, on mnist, mushroom, and gisette, all yields worse test accuracy compared to selective transmission. This confirms the intuition that not all unlabeled instances are useful. It also suggests the necessity of using a selective sampling strategy

on the client. Not only the communication costs can be saved, but also a better model might be learned. Moreover, it can be observed that uncertain and submod show significant improvements over rand. They often converge faster than rand and lead to better optimal hypotheses. On the contrary, selecting most certain instances is not beneficial. On ijcnn1 and skin, the accuracy decreases over time (the accuracy of certain on skin drops to under 80% at 4000 instances, and is not shown to better see the other results), showing that a bad client selection strategy can have negative impact on the performance of the server's model. On mnist and mushroom, submod further improves over uncertain, while uncertain is better for gisette and ijcnn1.

List of Publications and Significant Collaborations that resulted from your AOARD supported project:

a) Ian E.H. Yen, Chun-Fu Chang, Ting-Wei Lin, Shan-Wei Lin, Shou-De Lin, "Indexed Block Coordinate Descent for Large-Scale Linear Classification with Limited Memory", ACM SIGKDD 2013

b) Jui-Pin Wang, Yu-Chen Lu, Mi-Yen Yeh, Shou-De Lin, and Phillip B. Gibbons. "Communication-Efficient Distributed Multiple Reference Pattern Matching for M2M Systems". IEEE ICDM 2013.

c) Han Xiao, Shou-De Lin: "Learning Better while Sending Less: Communication-Efficient Online Semi-Supervised Learning in Client-Server Settings" in submission

Attachments: Publications a), b) and c) in the same zip file

DD882: As a separate document, please complete and sign the inventions disclosure form.

Important Note: If the work has been adequately described in refereed publications, submit an abstract as described above and refer the reader to your above List of Publications for details. If a full report needs to be written, then submission of a final report that is very similar to a full length journal article will be sufficient in most cases. This document may be as long or as short as needed to give a fair account of the work performed during the period of performance. There will be variations depending on the scope of the work. As such, there is no length or formatting constraints for the final report. Keep in mind the amount of funding you received relative to the amount of effort you put into the report. For example, do not submit a $300k report for $50k worth of funding; likewise, do not submit a $50k report for $300k worth of funding. Include as many charts and figures as required to explain the work.