IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 2, February 2016.

www.ijiset.com

ISSN 2348 – 7968

An Optimal Machine Learning Approach for Large-scale Applications

S.Rukshana[1], A.Narayana Rao[2]

[1]PG Scholar, Dept of CSE,Shree Institute of Technical education, Tirupati, AP, India

[2]Assoc. prof,HOD,Dept of CSE, Shree Institute of Technical education,Tirupati,AP, India

Abstract-Online learning is an important module of optimal machine learning techniques used in large-scale application developments. Feature selection is found in many applications to solve the problems raised by high dimensional data. Our goal is to develop an efficient prediction model through pruning the dataset by filtering the irrelevant and duplicates. It ensures the best performance in learning, interpretation, dimensioning .We propose online feature selection on the medical domain for predicting the treatment and side effects on the diseases. We collect the dataset from a medical blog (Medline).which is traversed between unsupervised, semi-supervised & the supervised. We introduce 2C & 3C classifications for feature selection, we train the system with the selected features and the result shows the treatments recommended by the medical experts in the online-datasets.

Index Terms: Feature selection, online learning, classification, large scale data mining

1. INTRODUCTION

FEATURE selection (FS) is an important topic in data mining and machine learning, and has been extensively studied for many years in literature. For classification, the objective of feature selection is to select a subset of relevant features for building effective prediction models. By removing irrelevant and redundant features, feature selection can *improve the performance* of prediction models by alleviating the effect of the curse of dimensionality, enhancing the generalization performance, *speeding up* the learning process, and improving the model interpretability. Feature selection has found applications in many domains, especially for the problems involved high dimensional data. Despite being studied extensively, most existing studies of feature selection are restricted to batch learning, which assumes that the feature selection task is conducted in an offline/batch learning fashion and all the features of training instances are given a priori. Such assumptions may not always hold for real-world applications in which training examples arrive in a sequential manner or it is *expensive t*o collect the full information of training data.

For example, in an *online spam email detection* system, training data usually arrive sequentially, making it difficult to deploy a regular batch feature selection technique in a timely, efficient, and scalable manner. Another example is feature selection in bioinformatics, where acquiring the entire set of features/ attributes for every training instance is expensive due to the high cost in conducting wet lab experiments. Most existing studies of online learning require accessing all the attributes/features of training instances.

Traditional approach is not always appropriate for real-world applications when data instances are of *high dimensionality* or it is *expensive* to acquire the full set of attributes/features.

Most existing studies of feature selection are restricted to batch learning, which assumes that the feature selection task is conducted in an offline/batch learning fashion and all the

features of training instances are given a priori. Such assumptions may not always hold for real-world applications in which training examples arrive in a sequential manner or it is expensive to collect the full information of training data. For example, in an online spam email detection system, training data usually arrive sequentially, making it difficult to deploy a regular batch feature selection technique in a timely, efficient, and scalable manner. Another example is feature selection in bioinformatics, where acquiring the entire set of features/ attributes for every training instance is expensive due to the high cost in conducting wet lab experiments.

Feature Selection has been studied extensively in the Literatures of data mining and machine learning. The existing FS algorithms generally can be grouped into three categories: *supervised, unsupervised, and semi-supervised FS.Supervised FS* selects features according to labeled training data. Based on different selection criterions and methodologies, the existing supervised FS methods can be further divided into three groups: *filter methods, wrapper methods, and embedded methods* approaches. Filter methods choose important features by measuring the correlation between individual features and output class labels, without involving any learning algorithm.

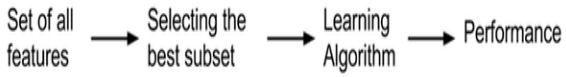


Fig 1: Filter method for selecting sub set

Wrapper methods rely on a predetermined learning algorithm to decide a subset of important features. Although wrapper methods generally tend to outperform filter methods, they are usually more computationally expensive than the filter methods.

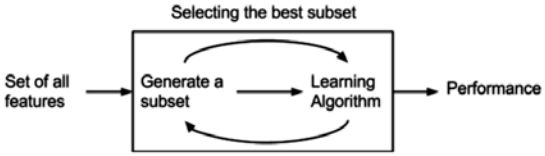


Fig 2: Wrapper method for selecting sub set

Embedded methods aim to integrate the feature selection process into the model training process. They are usually faster than the wrapper methods and able to provide suitable feature subset for the learning algorithm.

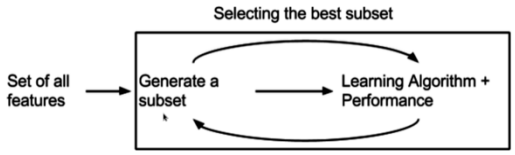


Fig 3: Embedded methods for selecting sub set

When there is *no label information* available, *unsupervised* feature selection attempts to select the important features which preserve the original data similarity or manifold structures. Some representative works include Laplacian Score [20], spectral feature selection.
We investigate three variants of budgeted learning, a setting in which the learner is allowed to access a limited number of attributes from training or test examples. In the "local budget" setting, where a constraint is imposed on the number of available attributes per training example, we design and analyze an efficient algorithm for learning linear predictors that actively samples the attributes of each training instance. Our analysis bounds the number of additional examples sufficient to compensate for the lack of full information on the training set. This result is complemented by a general lower bound for the easier "global budget" setting, where it is only the overall number of accessible training attributes that is being constrained. In the third, "prediction on a budget" setting, when the constraint is on the number of available attributes per test example, we show that there are cases in which there exists a linear predictor with zero error but it is statistically impossible to achieve arbitrary accuracy without full information on test examples..

We note that it is important to distinguish online feature selection addressed in this work from the previous studies of online streaming feature selection In those works, features are assumed to arrive one at a time while all the training instances are assumed to be available before the

ISSN 2348 – 7968

learning process starts, and their goal is to select a subset of features and train an appropriate model at each time step given the features observed so far. This differs significantly from our online learning setting where training instances arrive sequentially, a more natural scenario in real-world applications.

2. RELATED WORK

We propose a solution of online feature *selection* (O.F.S) with a classifier using short and scalable set features. The training set is taken online *and sequential.* The goal is *accurate prediction* for an instance with the help of minimum number of active-features. We address two different types of online feature selection tasks: O.F.S by learning with full inputs: where a learner is allowed to access all the features to decide the subset of active features,

2.1 A Simple Truncation Approach

We assume the learner is provided with full inputs of every training instance we first present a simple but non-effective algorithm that simply *truncates the features* with small weights. The failure of this simple algorithm motivates us to develop effective algorithms for OFS.

Input

•B: the number of selected features

2: **Initialization**

•$w_1 = 0$

3: **for** $t = 1,2\ldots\ldots T$ **do**

4: Receive x_t

5: Make prediction $\text{sgn}(x_{Tt}\ w_t)$

6: Receive y_t

7: **if** $y_t x_{Tt}\ w_t \leq 0$ **then**

8: $\tilde{w}_{t+1} = w_t + y_t x_t$

9: $w_{t+1} = T\ runcate(\ w_{t+1}, B)$

10: **else**

11: $w_{t+1} = w_t$

12: **end if**

13: **end for**

2.2 A Sparse Projection Approach

The online learner maintains a linear classifier has at most B nonzero elements. When a training instance is misclassified, the classifier is first updated by *online gradient descent* (OGD) and then projected to a L2 ball to ensure that the norm of the classifier is bounded.

If the resulting classifier has more than B nonzero elements, we will simply keep the B elements in,w_t the largest absolute weights O.F.S by learning with partial inputs: where only a limited number of features is allowed to be accessed for each instance by the learner. For computational efficiency we need to select a relatively small number of features for linear classification. We propose an "-greedy online feature selection approach with partial input information by employing a classical technique for making tradeoff between exploration and exploitation. In this approach, we will spend " of trials for exploration by randomly choosing B attributes from all d attributes, and the remaining 1" of trials on exploitation by choosing the B attributes for which classifier wt has nonzero values. We propose a greedy O.F.S algorithm with partial inputs and the performance is improved by selecting minimum features for the linear classifier.

This proposed greedy O.F.S algorithm with partial inputs would have these advantages: *online learning, accurate prediction, uses selected set of attributes and recommended for using datasets with high dimensional instances*.

3.ONLINE FEATURE SELECTION

Online feature selection can be done in four phases,

3.1.OFS: Learning

The online feature selection approach with partial input information by employing a classical technique for making tradeoff between exploration and exploitation: In this approach, we will spend ε of trials for exploration by randomly choosing B attributes from all d attributes, and the remaining 1−ε trials on exploitation by choosing the B attributes for which classifier $w_t$has non-zero values. Algorithm 1 shows the detailed steps of the proposed **OFS**P algorithm.

Algorithm 1: **BEGIN**

1.Collect abstracts
2.Pre-processing
3.Text extraction
4.If 2c **then**
5.Informative
6. 3c-feature extraction
7.**Else**
8.Non-informative
9. Goto preprocess step 3
10.**End if**
11.Update database
12.Calculate F-measure
13.Display R
14.END

3.2 Online collection of Medical Abstracts

The *Medline* is a *health blog* managed by the medical experts, it is a considered as a high knowledge base for online learning for Medical prediction systems. We collect the medical abstracts submitted by the medical experts in this blog, to extract the features to predict the *disease - treatment* relationships.

3.3 Information extraction(2c)

The first task identifies sentences from Medline published abstracts that talk about diseases and treatments. The task is similar to a scan of sentences contained in the abstract of an article in order to present to the user-only sentences that are identified as containing relevant information (disease treatment information). It identifies whether sentences are *informative*, i.e., containing information about diseases and treatments, or not.

3.4.Relation Identification(3c)

This task has a deeper semantic dimension and it is focused on identifying disease-treatment *relations* in the sentences already selected as being informative.The focus is to automatically identify which sentences contain information for the three semantic relations: Cure, Prevent, and Side Effect.

4.ARCHITECTURE DIAGRAM

The following architecture diagram shows how the feature selection will be done. Here first according to user,the data will be loaded then feature selection is performed in which informative and non- informative sentences will be resulted. In those informative sentences f-measure for each treatment and side effect is shown.

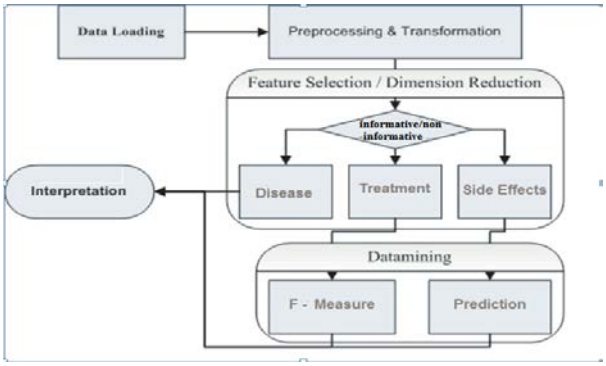
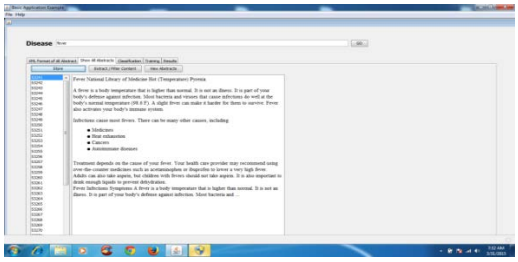


Fig 4 :Architecture of OFS

5. SCREEN SHOTS

When we enter a disease name we will get the details of it like treatment, side effects etc. First we will get all abstracts of the corresponding disease from the Medline blog in the XML form. Then those *XML abstracts* are filtered by our proposed algorithm as shown in the following screen shot.

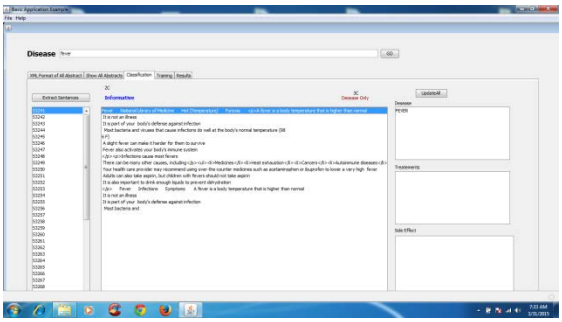


Screen1: Collecting XML abstracts from Medline

After collecting all abstracts will get the sentences and those sentences can be classified as 2c under this classification they will sub categorized as informative and non-informative.
Informative: The selected sentences are disease related and their details can be viewed in 3c.

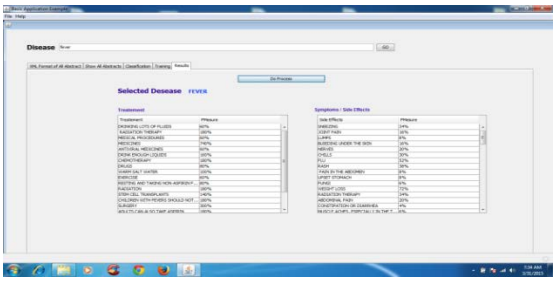
ISSN 2348 – 7968

Non-informative: The selected sentences are not related to user's disease. In this case the OFS process will be terminated.



Screen 2: 2C and 3C classification

And the following screen shot represents the treatment, side effects and their corresponding f-measures. It is the result of OFS (Online Feature Selection).These results are filtered by our new proposed algorithm so the performance speed will be high. These are only the references to the doctors and the people.



Screen 3: OFS results

1. http://www.ics.uci.edu/~mlearn/MLRepository.html
2. http://www.google.com/feature selection
3. http://www.wikipedia.org/wiki/Feature_selection#

6.PERFORMANCE EVALUATION:

The RAND algorithm suffered thehighest mistake rate for all cases. This again shows that it is important to learn the active features for the inputs and the weight vector. Second, OFSrandmade significantly more mistakes than OFSP for all the datasets, which validates the importance and efficacy of exploring the knowledge of the active features. Finally, we found that the proposed OFSP algorithms achieved the smallest mistake rates. This shows that the proposed OFSPtechnique is effective for learning the most informative features under the partial input situation.

6.1 ONLINE VS BATCH FEATURE SECTION METHODS:

When OFS and batch algorithm mRMR (minimum redundancy maximum relevance feature selection) are compared the data sets are divided into two equal sizes: the first part is used to select features by running FS algorithms (OFS and mRMR), and the second part is used to test the performance of selected features. To examine the efficacy of the selected features invariant to different classifiers, we adopt two types of widely used classifiers: (i) Online gradient descent (OGD) which is an online learning classifier, and (ii) K-nearest neighbor classifier (KNN), which is a batch learning classifier.

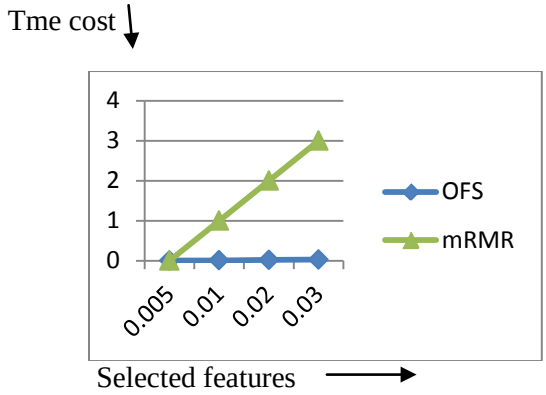


Fig 5S: OFS Vs mRMR

6.2 OFS VS SPARSE ONLINE LEARNING:

When the sparsity level is 0 (all the features are selected, for some text datasets, which adopt the bag-of-words representation, the features are already somewhat sparse), FOBOS(Forward Backward splitting)andOFS perform almost identically, which indicates the two methods

have very similar predictive performance for online learning. When the sparsity level increases, we observe that the proposed OFS algorithm significantly outperforms FOBOS. The FOBOS algorithm adopts the ℓ1 norm regularization based approach, in which the optimization task of FOBOS leads to the softthresholding operations in order to achieve the sparse solutions. In contrast, OFS have two important advantages:

(i) OFS can select the exact number of features specified by users, while FOBOS has to carefully tune the regularization parameter in order to achieve the desired sparsity level;

(ii) The soft-thresholding operations mayachieve different sparsity levels at different iterations during the online learning process.

6.3 OFS IN IMAGE CLASSIFICATION AND BIO INFOMATICS:

The effectiveness of feature selection in bioinformatics can be done by 2 particular problems, i.e., *population classification* with SNPs and *cancer classification* with microarray data. OFS proposed to find out which SNPs are significant in determining the population groups and to classify different populationsusing these relevant SNPs as the input features. A modified t-test ranking measure is applied on the problem of classifying populations from the *Hapmap* genotype data.

7.CONCLUSION

OFSaims to select a *small* and *fixed number* of featuresfor binary classification in an online learning fashion. Inparticular, we addressed two kinds of OFS tasks in twodifferent settings: 1) *OFS by learning with full inputs* of all the dimensions/attributes, and 2) *OFS by learning withpartial inputs* of the attributes. We presented a novel OFS algorithm to solve each of the OFS tasks and this algorithm is used to solve the problems raised by high dimensional data like *image classification* in computer vision and *microarray gene expression*analysis in bioinformatics.

7.REFERENCES

[1] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter, “Distribu-tional Word Clusters versus Words for Text Categorization,” J. Machine Learning Research, vol. 3, pp. 1183-1208, 2003.

[2] J. Bi, K.P. Bennett, M.J. Embrechts, C.M. Breneman, and M. Song,“Dimensionality Reduction via Sparse Support Vector Machines,”J. Machine Learning Research, vol. 3, pp. 1229-1243, 2003.

[3]G. Cavallanti, N. Cesa-Bianchi, and C. Gentile, “Tracking the Best Hyperplane with a Simple Budget Perceptron,Machine Learning, vol. 69, nos. 2-3, pp. 143-167, 2007.

[4]N. Cesa-Bianchi, S. Shalev-Shwartz, and O. Shamir, “Efficient Learning with Partially Observed Attributes,”J. Machine Learning Research, vol. 12, pp. 2857-2878, 2011.

[5] A.B. Chan, N. Vasconcelos, and G.R.G. Lanckriet, “Direct Convex Relaxations of Sparse SVM,” Proc. 24th Int’l Conf. Machine Learning (ICML ’07), pp. 145-153, 2007.

[6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online Passive-Aggressive Algorithms,” J. Machine Learning Research, vol. 7, pp. 551-585, 2006.

[7] K. Crammer, M. Dredze, and F. Pereira, “Exact ConvexConfidence-Weighted Learning,” Proc. Advances in Neural Informa- tion Processing Systems (NIPS ’08), pp. 345-352, 2008.

[8] K. Crammer, A. Kulesza, and M. Dredze, “Adaptive Regulariza- tion of Weight Vectors,” Proc. Advances in NIPS pp. 414-422, 2009.