ORNL/TM-2015/344

Machine Learning for Big Data: A Study to Understand Limits of Performance at Scale

Sreenivas R. Sukumar
Carlos E. Del-Castillo-Negrete

12/21/2015

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-2015/344

Computational Sciences and Engineering Division

Machine Learning for Big Data: A Study to Understand Limits of Performance at Scale

Authors
Sreenivas R. Sukumar
Carlos E. Del-Castillo-Negrete

Date Published: 12/31/2015

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

ACRONYMS

ORNL	Oak Ridge National Laboratory
U.S.	United States
FBI	Federal Bureau of Investigation
SVM	Support Vector Machine
BOW	Bag of Words
CH	Color Histogram
CORR	Color Auto-Correlogram
EDH	Edge Detection Histogram
WT	Wavelet Texture
CM55	Color Moments
BOW	Bag of Words

ABSTRACT

This report aims to empirically understand the limits of machine learning when applied to Big Data. We observe that recent innovations in being able to collect, access, organize, integrate, and query massive amounts of data from a wide variety of data sources have brought statistical data mining and machine learning under more scrutiny, evaluation and application for gleaning insights from the data than ever before. Much is expected from algorithms without understanding their limitations at scale while dealing with massive datasets. In that context, we pose and address the following questions – How does a machine learning algorithm perform on measures such as accuracy and execution time with increasing sample size and feature dimensionality? Does training with more samples guarantee better accuracy? How many features to compute for a given problem? Do more features guarantee better accuracy? Do efforts to derive and calculate more features and train on larger samples worth the effort? As problems become more complex and traditional binary classification algorithms are replaced with multi-task, multi-class categorization algorithms – do parallel learners perform better? What happens to the accuracy of the learning algorithm when trained to categorize multiple classes within the same feature space? Towards finding answers to these questions, we describe the design of an empirical study and present the results. We conclude with the following observations – (i) accuracy of the learning algorithm increases with increasing sample size but saturates at a point, beyond which more samples do not contribute to better accuracy/learning, (ii) the richness of the feature space dictates performance - both accuracy and training time, (iii) increased dimensionality often reflected in better performance (higher accuracy in spite of longer training times) but the improvements are not commensurate the efforts for feature computation and training and (iv) accuracy of the learning algorithms drop significantly with multi-class learners training on the same feature matrix and (v) learning algorithms perform well when categories in labeled data are independent (i.e., no relationship or hierarchy exists among categories).

1. INTRODUCTION

Two Big Data projects in healthcare and law enforcement at the Oak Ridge National Laboratory (ORNL) provided the opportunity to survey the state-of-the-practice and apply state-of-the-art techniques to understand the gaps and challenges of machine learning at scale. We introduce the two projects and abstract the machine learning problem underlying the two use-cases below.

1.1 MOTIVATING USE-CASES

1.1.1 Healthcare

In 2011, United States (U.S.) Department of Energy's Oak Ridge National Laboratory (ORNL) and the Centers for Medicaid and Medicaid Services under the Department of Human Health Services collaborated via an inter-agency agreement to explore data-science and knowledge discovery opportunities in healthcare. At that time, ORNL possessed some the world's best computing resources and the Department of Human Health Services was hosting and processing the world's largest digital archive of healthcare transactions. The challenge for the inter-agency partnership was to leverage 'Big Health Data' towards smarter healthcare by discovering opportunities for better policy, quality and integrity. In other words, the challenge was to transform claims-oriented data to actionable knowledge for improving the quality of healthcare (cost-care optimization problem), detecting and preventing fraud, waste and abuse (data mining problem), and finding data-driven evidence (searching for trends, patterns and correlations) for aggressive pro-active policy decisions.

1.1.2 National Security and Law Enforcement

In 2012, after the Boston Marathon Bombing, the Federal Bureau of Investigation (FBI) was quickly inundated with multiple terabytes of videos, photos, tips, and social media data of the bombing event. Analyzing this data required a team of agents to manually review for clues, ultimately taking four days to identify a suspect. That four-day window provided the suspects with additional opportunities to either escape or commit additional crimes. Law enforcement agencies face a similar challenge of sifting through evidence in the fight against human trafficking. Evidence of crimes against children average four terabytes of data for each apprehended perpetrator - the hard drives include massive collections of videos and images of children being raped along with e-mail, social media connections to potential victims, and potential links to trafficking networks. Due to the limitations of existing forensic tools, there is a six-month backlog in analyzing a hard drive. At ORNL, we evaluated the art-of-the-possible with machine learning to provide solutions to the image triaging problem by proposing a system that can automatically describe or tag image content. We conducted a feasibility study by scraping millions of images from the web (a few of them pre-labeled or tagged) and using state-of-the-art learning methods to automatically come up with a conceptual description of the image.

1.2 PROBLEM STATEMENT

Although healthcare and national security appear as tangential application domains, both the use cases shared a similar formulation of the machine learning problem statement: Given a matrix **M** of data points ***x*** with **N** samples, along ***d*** feature dimensions and ***k*** categories or classes, find a function ***f*** that can predict categories for new samples of ***x***. Translating it into the respective domains, given longitudinal history of several patients, predict future needs – diagnoses, procedures and thereby cost for current and future patients. For the image triaging use-case, based on examples of tagged/labelled images, predict word association for new images. The size **N**, ***d*** and ***k*** (the volume, velocity and variety) are comparable in both applications - millions of patients making billions of claims in a year along thousands of possible diagnoses and millions of users on photo-sharing sites uploading billions of pictures with thousands of word tags. When posed with such Big Data, where the data sizes are huge and data scientists tasked with building predictive models, be it for fraud detection/prevention or for recommending healthcare products and services or recommend conceptually and perceptually similar advertisements – the state of the practice would involve defining a feature space of ***d*** dimensions that will categorize ***k*** classes to a desired level of accuracy, precision and recall. Today, data scientists do not have approximations or a deterministic theoretical bound on the number of dimensions ***d*** required for a feature matrix **M.** There are no guarantees on the expected accuracy and precision and no automated way to design feature spaces given raw datasets. A lot is expected from the "art" of deriving features from data using subject matter/domain expertise – which can be daunting given the velocity, variety and volume aspects of Big Data.

Several questions are still unanswered – How do machine learning algorithms perform on measures such as accuracy, precision and execution time with increasing sample size and feature dimensionality (i.e. increasing volume and variety)? Does training with more samples guarantee better accuracy? How many features to compute for a given problem? Do more features guarantee better accuracy? Does the investment to derive and calculate more features and train on larger samples worth the effort? As problems become more complex and traditional binary classification algorithms are replaced with multi-task, multi-class categorization algorithms – do parallel learners on the matrix **M** perform better? What happens to the accuracy of the learning algorithm when trained to categorize multiple classes within the same feature space? Does increase in model complexity (number of features ***d*** and parameters of the predictive function ***f***) help us understand the data better? Does increase in model complexity provide better accuracy, precision and recall? How many different models can an algorithm learn simultaneously? How to scale up/automate the feature engineering process? How can we recommend choice of analysis/classification algorithms based on the data characteristics? How do existing machine learning methods evolve to increasing samples, dimensionality and categories over time – as new samples stream in and archives available for training constantly increase in size? Our goal is not to provide answers to all of the aforementioned questions, but to report results from a study that reveals open challenges and opportunities for future machine learning research through an empirical understanding of accuracy, precision, recall and computational performance of a machine learning algorithm when applied to Big Data.

2. DESIGN OF THE EXPERIMENT

We focus on the automated image-tagging use case for our experiments. Image recognition and classification has been area of central importance in computer vision with applications from optical character recognition to smart cars. However, most of the success has been on relatively small scales when compared to the scale at which the human visual system performs the task of image recognition. Even state of the art machine learning techniques fail to reach the accuracy and efficiency of the human visual system. The complexity of the image classification problem captures the intrinsic characteristics of Big Data. First, the breadth of the semantic space of possible categories and concepts associated with an image (a dictionary is in the order of 100,000 words) makes large-scale image classification extremely challenging. Not only this, there are many ways to describe an image into a feature vector for training a model, and it isn't always clear which feature vectors lead to the best predictive models. It is impossible to manually engineer features for image recognition. Lastly, image data sets are growing by the day and are massive, exposing memory requirements for near real-time runtime processing. In spite of these high expectations we have chosen the image-analysis use case for the following two reasons - (i) the maturity of the image processing literature to transform images into informative feature-vector representations and (ii) the availability of ground truth and therefore the ability to visually/numerically validate the output of the machine learning algorithms.

2.1 DATASET - THE NUS-WIDE IMAGE DATA SET

We have conducted our study of the **N-*d*-*k*** scalability problem using the NUS-WIDE Real-World Web Image Database [9]. This image set consists of 269,648 images that are represented in six different types of low-level feature vectors (listed in Table 1). These feature vectors encode the image content of each image using several perceptual heuristics. One can think of the heuristics as a mathematical way to extract and quantify salient content within the image. The NUS-WIDE dataset is available as a feature matrix where features have already been calculated and compiled into usable data ready for training machine learning algorithms. Each image is also associated with one or several ground-truth concepts from a list of 81 different classes. The ground-truth concepts span a wide range of different concepts, from the very broad terms, such as *animal* or *sky*, to the more specific ones, such as *cat* or *car*.

2.2 FEATURE SPACE

State of the art image processing algorithms are able to encapsulate color, edge, texture, and other content of an image into feature vectors. The color, edge, texture, etc. serve as different heuristics of the feature space that can be used to train the image-to-word classifiers. More specifically, we leverage six different low-level feature vectors to train our classifiers. Table 1 listed the different image feature sets used in this experiment. The different feature heuristics (color histogram, color auto-correlogram, bag-of-words, etc.) emulate different subject matter experts creating a feature space for the image data. We note that each heuristic has a different sized feature vector that enables the association between the dimensionality of the feature vector and the efficiency of learning. In addition, by combining more than one heuristic, we will also study the performance of the machine learning algorithm as new features are incrementally made available for training.

Table 1: Name, abbreviation, dimensionality and description of feature vectors in NUS-WIDE image data set.

Feature Vector	Size (d)	Image Content
Color Histogram (CH)	64	Basic color content
Color Auto-Correlogram (CORR)	144	Color distributions and spatial correlations of pairs of colors
Edge Detection Histogram (EDH)	74	Distribution and directions of edges
Wavelet Texture (WT)	128	Multi-resolution texture analysis
Color Moments (CM55)	225	Color distributions according to the first three color moments
Bag of Words (BOW)	500	Treats image features as words and categorizes according to occurrence counts of a vocabulary of 500 local image features.

2.3 CLASSIFICATION ALGORITHM

After a detailed survey of different machine learning algorithms [4, 7] and off-the-shelf scalable and open-source implementations, we focused on the linear Support Vector Machine (SVM) method originally developed by Vapnik [1]. SVMs work by finding the maximum separating hyperplane between two classes in the ***d***-dimensional feature space in which images are represented. To classify new images, the SVM simply calculates what side of the hyperplane a feature vector maps to and classifies it accordingly. The SVM model is a binary classifier and performs multi-class classification by breaking the problem down into several different instances of binary classification. In particular, to train a ***k***-class classifier, the SVM method breaks down the problem into ***k*** binary classification problems of one class versus all the other classes. For each new image to classify the SVM obtains ***k*** scores from the ***k*** different one-vs.-all models and labels the given test image as a member of the class for which the corresponding one-vs.-all model achieves the highest score. We chose SVM over the plethora of methods available as open-source software because of the linear nature of the algorithm and also the ability to train several predictive functions in parallel. We leveraged the LIBLINEAR implementation of linear SVMs for our experiments [2]. We worked with the linear SVM models that LIBLINEAR implements using very efficient techniques to scale for large **N** and ***d***. Previous studies in large scale image classification have also used the LIBLINEAR libraries; see for example [3].

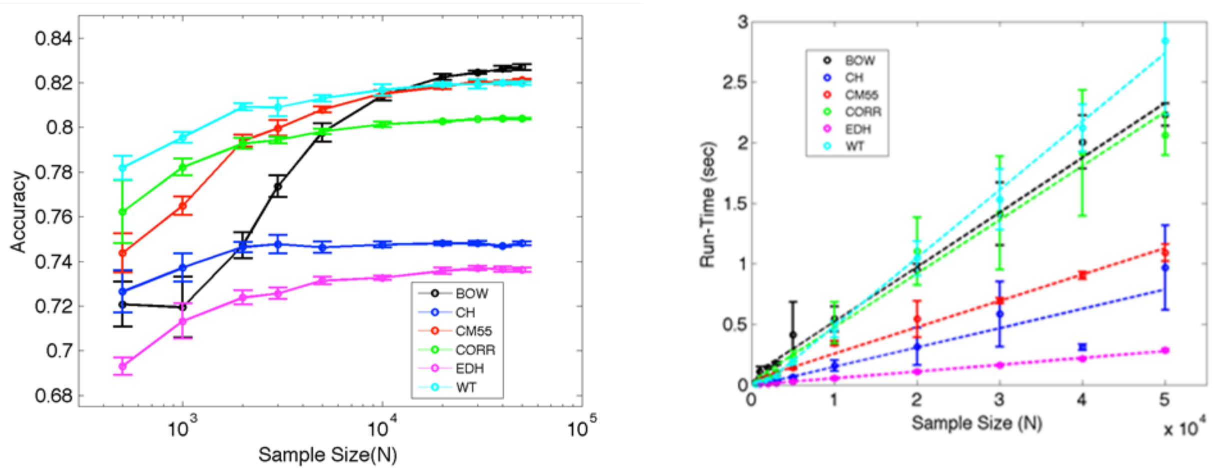
2.4 EXPERIMENTS

We developed a C-library of wrapper functions that construct different instances of classification problems with varying **N**-***d***-***k*** parameters and record the performance and accuracy of training a linear SVM classifier on the subsets of the feature-space created from the 269,648 images in the NUS dataset. Our library was built on top of the LIBLINEAR library leveraging the training and testing functions on the problem instances constructed from the NUS-WIDE data set. We automated the process of constructing and testing these data sets to iterate over a varying amount of image sample sizes (**N**), feature vectors (***d***), and number of classes (***k***). Furthermore for any given **N**-***d***-***k*** instance, we trained the same model five times with five different samples of images to make sure execution times are reliable and not affected by data-hotspots (e.g. i/o slow down, memory leaks, etc.). In addition to recording runtime and accuracy, we also computed the confusion matrix for each model when applied to the test set. The confusion matrix provides a compact visual representation of the accuracy of a multi-class classifier. Training and test sets were always divided in a 0.8-train, 0.2-test ratio. All tests were run on a local workstation with 16GB of memory. We have assumed that the feature matrix **M** is a lot smaller than the available memory on the machine and the training happens in memory (with minimal disk interaction).

3. OBSERVATIONS

3.1 RELATIONSIP BETWEEN TRAINING TIME AND INCREASE IN SAMPLES, DIMENSIONALITY AND CATEGORIES

We begin this section with a discussion on how an off-the-shelf machine learning algorithm performs on a Big Data problem – with increasing sample size (**N**), feature dimensionality (***d***) and categories (number of classes) (***k***). Our observations are presented in the Figures 1, 2 and 3 below for increasing size, dimensionality and categories respectively. As one would expect, using more samples to learn produced better accuracy while still remaining relevant with run-times and more features contributed to better accuracy at the cost of slightly longer training times. However, the learning algorithm does not perform to expectations when trained to learn multiple classes/labels/categories. The algorithm takes at least an order of magnitude longer to train (three orders in some cases) for the multi-class classification compared to the binary classification problem. In Figure 1, we study the performance of the linear SVM by incrementally increasing the size of the training set (**N**) presented to the algorithm, training the SVM algorithm for that sample size and then comparing the performance of the trained SVM models with different sized training sets on the same classification problem. In this case it was to classify images from two categories 'sky' and 'person'. We see the learning curve of the SVM algorithm to increasing data-size in Figure 1a below. We have plotted the learning curve for different feature heuristics also in the same figure.

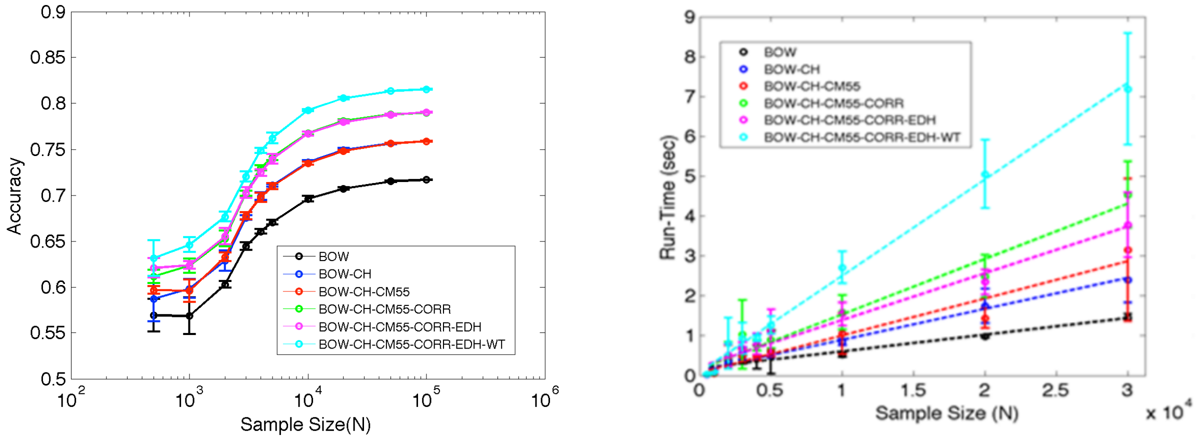


(a) (b)

Fig 1. Performance of the SVM learning algorithm for increasing sample size N on (a) accuracy and (b) time required for training. We observe that the SVM model scales well with the size N of the training set – i.e., adding more images to the training set definitely increases the accuracy of prediction.

We observe that there is a steep learning curve for all the models trained and there is a limit to how much the model can learn – i.e. learning saturates fast to the point of diminishing returns. Furthermore, the overall shape of the learning curve irrespective of the feature heuristic guides us to the conclusion that, while increasing **N** does increase the accuracy of the model trained regardless of feature vectors used, some feature heuristics can take ten times longer to train at scale compared to others. Accuracy of classification improves with increasing size of the training set and the efficiency of the feature vector determines the slope of the learning curve (i.e. some descriptors are more efficient than others). However, how much you can learn appears to have an empirical limit. The improvement in accuracy with increasing samples saturates and appears to obey the law of diminishing returns [5]. This result reveals that more data does not always mean better classification accuracy. More computational resources spent on training and building a classification model with Big Data will not guarantee a proportional or significant increase in the quality of classification.

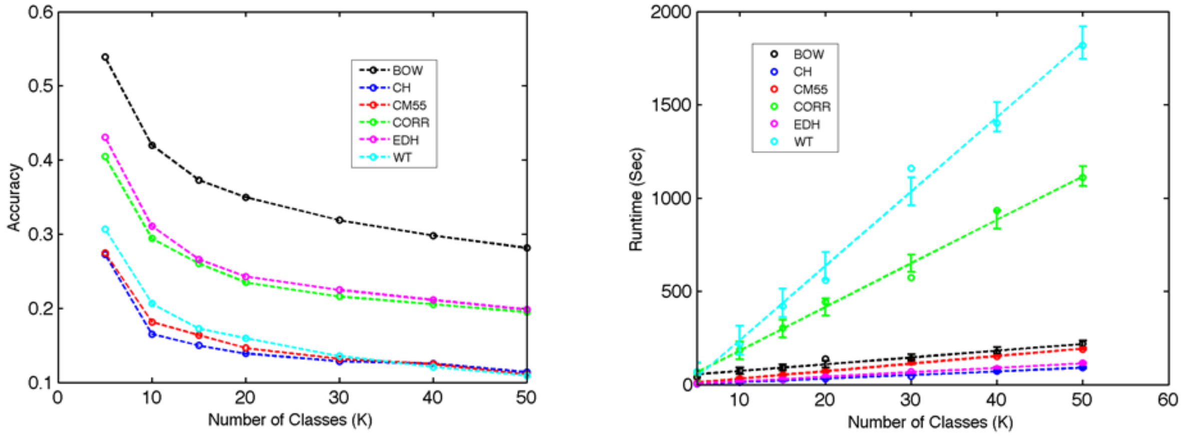
To study the effect of scaling-up in ***d*** we increased the size of the feature space by incorporating different combinations of feature vectors to train the SVM model. The different sets of feature vectors were simply concatenated together and the SVM model was tested using these concatenated feature vectors. Figure 2 is the results of training a SVM to classify '*animal*' vs. '*person*'. Starting with the BOW feature vector, we appended each feature vector incrementally and recorded the performance of the SVM model trained with these new feature vectors at different sample sizes. Despite the relative simplicity of the concatenation approach, our method of combining different feature vectors showed promising results in increasing the descriptive power of the SVM model. The trends with the timing results were still within acceptable near-real time requirements of most applications although increasing size of the feature vector translated to increasing training time.



(a) (b)

Fig 2. Performance of the learning algorithm for increasing feature dimensionality *d* on accuracy and learning time.

To study the ***k***-scalability of the linear SVM classifier we fixed ***N*** and ***d*** and successively increased the number of classes in the training set from ***k***=5 to ***k***=50. For each ***k***-classification problem, we chose the most populated classes (i.e., most represented) in the data set and used 130,000 images to train the linear SVM model. The results for different feature vectors are summarized in Figure 3. We notice the rapid decrease in accuracy with increasing ***k***. Of note also is how different feature vectors perform dramatically different as ***k*** increases. The bag-of-words feature heuristic is clearly the best in terms of accuracy and still very efficient in its runtime. WT quickly degrades to mediocre accuracy and blows up on training time for high ***k***. The experimental results from Figures 1, 2 and 3 motivated further experiments that we discuss below.

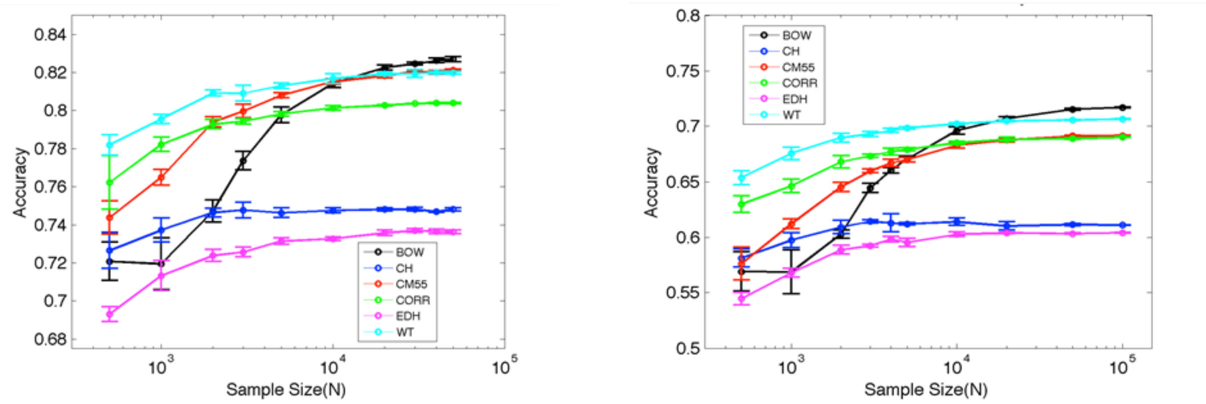


(a) (b)

Fig 3. Performance of the learning algorithm for increasing number of categories (*k)* on accuracy and learning time.

3.2 SIGNIFICANCE OF FEATURE SPACE TO MULTI-CATEGORY CLASSIFICATION

Figure 4a shows the result of a binary classifier (***k***=2) and Figure 4b is the result of classifying over 3 labels (***k***=3) for all the 6 aforementioned feature heuristics. The plots show the significance of a feature space to classification in two aspects – accuracy and rate of learning. We see that the wavelet texture based heuristics perform the best for smaller sample sizes (Figure 4a), but as more samples are provided for training the bag-of-words heuristic eventually outperforms all the other feature spaces on accuracy. This implies that the quality of features influence how many training samples a classifier needs, how quickly can a predictive model be trained and how much effort to spend on training. Comparing the plots Figure 4a and 4b, the trends look similar. But, the accuracy for each feature-space has gone down by approximately 12%. This is because the discriminatory power of a feature space for ***k****=2* problem is insufficient (significantly less) for the ***k****=3* problem. This exposes the need for creative feature engineering - particularly before extending the application of a feature space to incremental newer problems. Conversely, this result also exposes the shortcoming with SVMs to incrementally improve the "expressiveness" (through increased parameter and model complexity) of the predictive model.

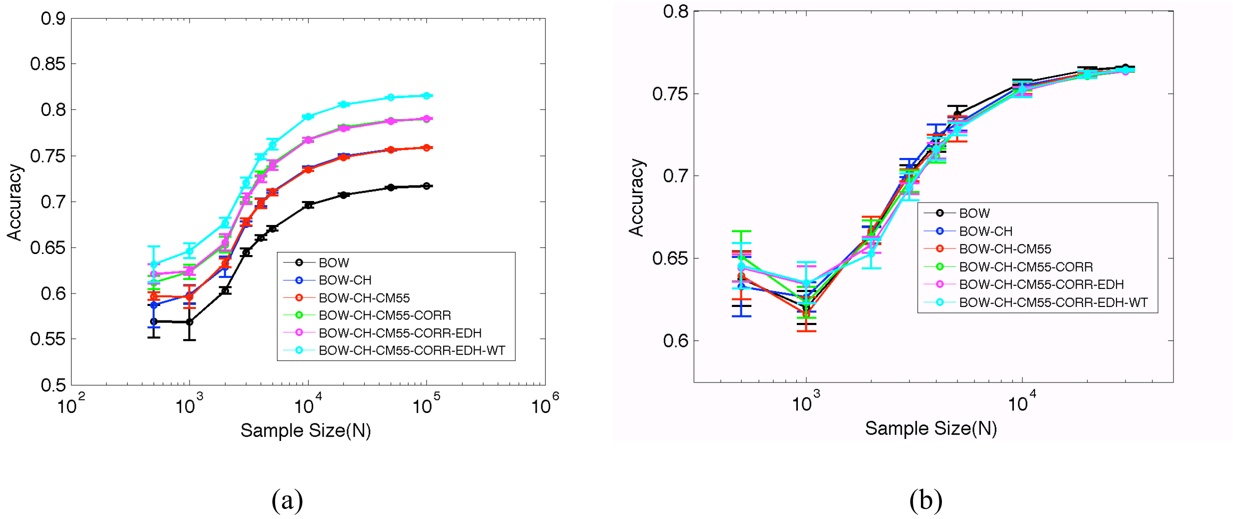


(a) (b)

Fig 4. Understanding the significance of the richness of a feature set to increasing N and k. (a) k=2 (b) k=3. The graphs confirm that the quality of the feature space dictates the performance - both accuracy and training time.

3.3 IMPACT OF FEATURE DIMENSIONALITY ON A MUTLI-CATEGORY MULTI-TASK CLASSIFICATION PROBLEM

Figure 5a and 5b are the result of training and classifying using each feature space by incrementally concatenating feature vectors for two parallel learners – '*animal* vs. *person*' and '*cloud* vs. *sky*'. For both cases the training time consistently increased with concatenated feature vectors. While analyzing results of other trained multi-task classifiers in the ensemble, we observed that in the majority of classification problems, concatenating more feature vectors leads to an improved accuracy for the binary classification problem and in some cases concatenating feature vectors had no statistically noticeable effect on accuracy of the trained model. An example of one such result is presented in Figure 5b. Further investigation revealed that semantically distant classes (e.g. '*sky'* and '*plants'*) benefited from the concatenated feature vector while semantically close categories did not (e.g. '*cloud'* and '*sky'*).



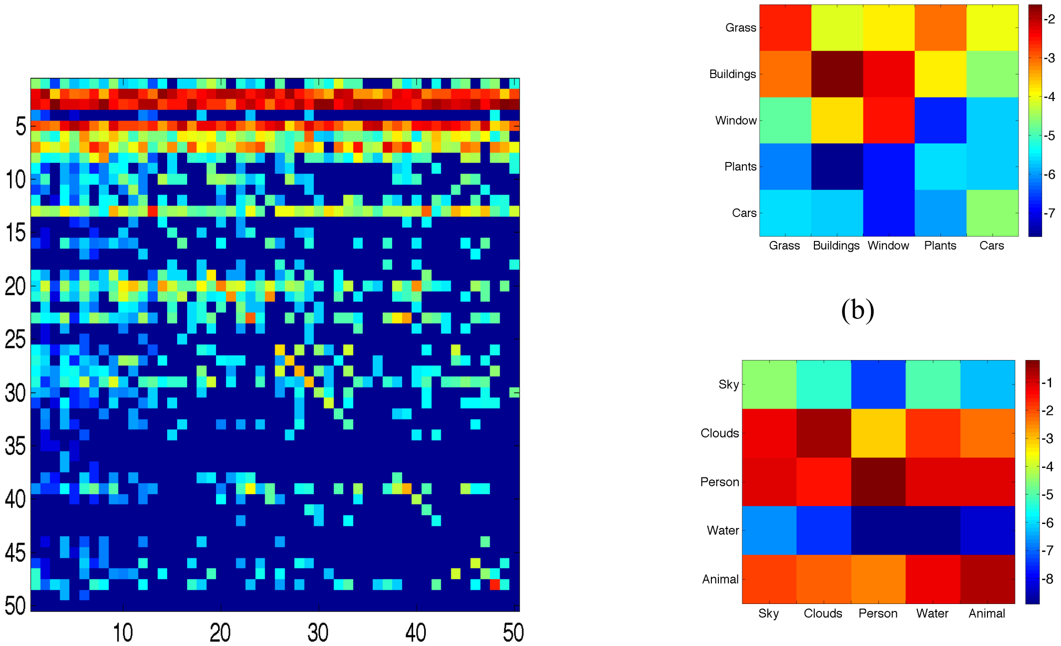
(a) (b)

Fig 5. Understanding the significance of the feature dimensionality during multi-category training using multi-tasking parallel learners. (a) '*Animal* vs. *person*' and (b) '*Cloud* vs. *sky*'. With increasing dimensionality, we are observing that the classifier requires more samples to achieve a desired level of accuracy.

3.4 IMPACT ON ALGORITHM PERFORMANCE WHEN CATEGORIES ARE NOT INDEPENDENT

In the previous section, we discovered that relationship among categories can mislead a learning algorithm. In this section, we attempt to understand such an impact on the SVM learning model in the high-***k*** classification problems using the confusion matrix in Figure 6. The confusion matrix allows us to identify and analyze the difficult classes/categories during the training phase of the SVM. Figure 6 shows the confusion matrix for the SVM model learned using the BOW feature vector. We are able to observe that highly populated classes introduce sample bias. The SVM learns broad concepts, such as '*person*' or '*animal*' better than specific terms with limited examples.

Also, further investigation indicated that, the SVM learning algorithm favors classifying towards a broader category than a more specific one when categories are organized in a hierarchy. For our experiments, we used WordNet [6] to quantify the relatedness between categories. The proximity (degree of separation) in the WordNet tree was considered as the ground truth for conceptual relatedness among categories. We were able to conclude with statistical significance that the SVM classifier had difficulty distinguishing semantically close concepts compared to semantically distant ones. One example of this observation is presented in Figure 6. The top right inset of the confusion matrix in Figure 6 reveals there is rarely any confusion classifying '*plants*' from '*cars*', while distinguishing '*buildings*' from '*windows*' is more difficult. A similar example is also seen with another example with '*sky*', '*clouds*' and '*animal*' in the bottom inset. The more hierarchy we introduced in the categories (complex semantic structure in this image example case), the worse the learning algorithm performed. We see performance differences in being able to classify '*buildings'* as buildings and '*cars*' as cars. We attribute this to the explanation that the variety of *cars* is significantly larger than the variety in '*buildings'* and is the artifact of not being able to sample a large enough dataset to learn the variety.



(b)

(a) (c)

Fig 6. Understanding the ability of the machine learning algorithm using the confusion matrix over the *k*-categories (*k*=50) when they are not independent of each other and instead have hierarchies of organization. The red pixels at the top of the matrix correspond to classes with the largest training set. The learning algorithm seems to be defaulting towards the most populated classes in the data set, indicating that the unbalanced nature of the data set skews the SVM. (a) Semantic relationship example #1 with '*plant*' and '*cars*' vs '*buildings*' and '*windows*' (b) Semantic relationship example #2 with '*sky*' and '*clouds*' vs. '*sky*' and '*animal*'.

4. CONCLUSIONS AND FUTURE WORK

This study was aimed at understanding how machine learning algorithms scale to Big Data. We developed software to explore the scalability of the SVM classification algorithm and applied it to the image recognition problem. Our study provided valuable insights into not only how the SVM algorithm scales up and where it falls short, but also provides us insight into opportunities to create smarter and more efficient scalable classification algorithms that are fine-tuned for Big Data challenges.

We framed three main issues of scaling up machine learning as the **N-*d*-*k*** scalability problem where **N** refers to the number of sample instances used to train a particular model; ***d*** refers to the size (dimensionality) of the feature space used to encode an image; ***k*** refers to the number of classes in the classification/prediction problem. Based on the experiments, we concluded with the following observations – (i) accuracy of the learning algorithm increases with increasing sample size but saturates at a point, beyond which more samples do not contribute to better accuracy/learning, (ii) the richness of the feature space dictates the performance - both accuracy and training time, (iii) increased dimensionality often reflected in better performance (higher accuracy in spite of longer training times) but the improvements are not commensurate the efforts for feature computation and training and (iv) accuracy of the learning algorithms drop significantly with multi-class learners training on the same feature matrix and (v) learning algorithms perform well when categories in labeled data are independent (i.e., no relationship or hierarchy exists among categories).

Philosophically speaking, the foundations of traditional statistical machine learning have worked under the assumption that training data is scarce. With Big Data, the data is abundant and the bottleneck is the computation time. Even if data scarcity is still a problem with Big Data, our experiments have exposed several theoretical gaps to address. We have emphasized the need for developing newer scalable machine learning algorithms. We showed that increasing the training set size cannot improve classification errors indefinitely. This shortcoming we believe is because of taking an algorithm derived with statistical sincerity for smaller feature spaces and assuming that the same theory is robust to scale for requirements on Big Data problems. Our observations make the argument for a robust theoretical framework specific to Big Data problems – approximate reduced-order theoretical bounds, incremental feature engineering algorithms and more expressive family of predictive functions,.

Towards that goal, we motivate future work along the following themes:

1. *Theoretical approximate bounds*: We envision the theory will be an extension of the Bayes Limit [8] for binary classifiers to multi-category classifiers. For example, the rule of thumb is to use $\mathbf{N} > \boldsymbol{2}^{\boldsymbol{d}}$ samples to train a binary classifier training. Scaling up using traditional methods for the ***k***-category problem will require $\mathbf{N} > \boldsymbol{k}^{\boldsymbol{d}}$ samples – exorbitant both from the perspective of data collection and computational time for training. Reduced-order sub-optimal stochastic learning methods that are emerging as alternatives to traditional statistics-based learning need theoretical guidance and validity. In other words, given a feature matrix **M**, and an expectation of a time required to train the model or a requirement on desired accuracy, we should be able to derive a theoretical reduced-order formula to approximate how many examples are needed for the learning problem to meet both accuracy and computational time constraints

2. *Incremental feature engineering*: We are convinced that the quality of the feature space determines the performance of the algorithm and any learning algorithm is only as good as the feature space it is presented with. Unfortunately, feature engineering is the most time-consuming expensive (often requiring manual domain expert's help) aspect of training a machine learning algorithm. Furthermore, the engineering hurdles of handling Big Data, poses a hurdle to the manual feature extraction process. We need dynamic and intelligent ways of generating feature spaces for Big Data problems. Particularly, features that are robust to increasing complexity of the learning problem (e.g. adding new category of labels), self-correcting to sample bias, and also guiding the rate of learning by revealing characteristics of data conditional upon the probability distribution of the active working feature set.

3. *Expressive family of predictive functions*: We need to derive algorithms that are able to learn the structure (ontology) of the ***k***-category space in addition to being able to predict the categories based on the feature attributes. These algorithms will require a family of predictive functions that can seamlessly scale for complexity, size and training time constraints.

5. ACKNOWLEDGEMENTS

This research was partly supported by the Department of Energy Summer Undergraduate Laboratory Internship Program in 2014. The research was also supported by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy.

REFERENCES

1. Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
2. Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research, 9, 1871-1874.
3. Deng, J., Berg, A. C., Li, K., & Fei-Fei, L. (2010). What does classifying more than 10,000 image categories tell us?. In Computer Vision–ECCV 2010 (pp. 71-84). Springer Berlin Heidelberg.
4. Alpaydin, E. (2014). Introduction to machine learning. MIT press.
5. Samuelson, P. A. (1954). The pure theory of public expenditure. The review of economics and statistics, 387-389.
6. Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.
7. Bekkerman, R., Bilenko, M., & Langford, J. (Eds.). (2011). Scaling up machine learning: Parallel and distributed approaches. Cambridge University Press
8. Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine learning, 29(2-3), 103-130.
9. Chua, T. S., Tang, J., Hong, R., Li, H., Luo, Z., & Zheng, Y. (2009, July). NUS-WIDE: a real-world web image database from National University of Singapore. In Proceedings of the ACM international conference on image and video retrieval (p. 48). ACM.