Master Thesis Proposal

# Applications of Scalable Machine Learning Models

MINGZHE XU
*Northeastern University*
April 20, 2016

## Contents

# 1 Introduction

Machine learning is a method to enable computers to learn without explicitly being programmed[20]. In the late 1990s, computer scientists started to train computers to build models given the advance of digitalization and cheaper computing power. With the vast volume and increasing complexity of data, as well as accessible and affordable storage in the past decades, interests in machine learning surged unprecedentedly and applications became almost ubiquitous, which make it possible to quickly and automatically produce models that can analyze complex data and deliver faster, more accurate results. Thomas H. Davenport once wrote in The Wall Street Journal that "humans can typically create one or two good models a week; machine learning can create thousands of models a week"[27]. Precisely. Machine learning instantaneously presents high-value predictions that can guide better decisions and smarter actions in real time without any human intervention.

## 1.1 Applications of Machine Learning

The world now is swimming in machine learning products more than ever. In a variety of industries and analytical settings where access to troves of useful and reliable data is provided, machine learning is being frequently applied to explore useful information, generate accurate prediction, and directly or indirectly increase revenues. Online retailers can recommend products closely related to customers' interests. Amazon personalized recommendations, which base on customers' order history, items in carts, items rated or liked, and products similar customers viewed and purchased, provide heavily customized browsing experience for users. This personalized marketing has generated lots of revenue. According to a survey conducted in 2015, 68.4% of all revenue of the online retailer was from the recommendation system[28]. Netflix also provide individualized recommendations that greatly improved user experience and more than half of the viewing is from recommendation. Similarly, YouTube's Watch Next is also an embodiment of personalized recommendation. What's more, machine learning also pops up in web search results. Search engine is the lifeblood to Google, and personalized search results as well as customized ads display are core to the click through rate of the results. They are based on the combinations of queries, ads, results, and users' browsing history. In addition to Google, Facebook's News Feed, LinkedIn's Posts also ranks in

a customized order. In banking system, credit scoring, fraud detection and next-best offers are all built on machine-learning algorithms[2]. The recent popular computer program AlphaGo beats professional human Go players after learning from a database of around 30 million moves[22]. In speech and image recognition, machine-learning techniques are also widely used.

In the meantime, there are constraints. With the exceptional increase in computing power, storage capacity and network bandwidth in the past decade, datasets are growing fast in fields such as search engine, bioinformatics, IT-security, speech/image recognition, or email record, to name but a few. The growth of data size leaves computational methods the only viable way of dealing with data. However, it poses new challenges to machine learning methods. Google processes around 100 PB per day on 3 million servers; YouTube has 300PB storage and around 4 billion views per day[25]. The Big Data industry roughly process 2.5 quintillion bytes of data every day[26]. And the number is not still. Looking forward, in four years there will be 40 zetta-bytes of data. Therefore, how to make the learning process more efficient is vital to those companies. With at least 3.5 billion requests coming in everyday, what algorithms can make faster and more accurate response? What models can survive the training process on petabytes of data?

## 1.2 Scalability of Machine Learning

This explosion of data tuned widespread attention to scalability, especially learning large-scale data. Scalable machine learning involves processing paradigms, statistical analysis, algorithms for data streams, and large scale convex optimization, etc. It is an integration of system, statistics, data mining and machine learning. Scalability is one of the key concepts in Big Data. Yet far beyond Big Data, scalable or large scale was a buzzword in machine learning ever since there were large amount of data such as dealing with text document or in bioinformatics. Datasets with large number of features, samples, or even when the data can't fit into the memory, solutions are needed to enable learning and processing. Currently, there are some solutions on different levels. In memory, we can intellectually swap between memory or disk. On the algorithmic level, we can use online algorithm which can constantly learn and predict as the data flows[6]. Also, faster and optimized algorithms are needed. Parallel algorithms are vital to higher efficiency. When data cannot fit into memory of a single machine,

distributed algorithms can tackle this problem.

### 1.3 Thesis Objectives

There are two core purposes for this thesis course. First is learning and studying scalable machine learning theories. Laying a firm theoretical foundation for conducting practical experiments on large scale machine learning is crucial. The second purpose is to apply the concepts into solving real world problems with large data so as to put the theories learned into practice.

## 2 Theoretical Foundation

Scalable machine learning is constructed with several key subjects. First is machine learning models that can be scalable. Second, efficient learning algorithms are key to a better prediction, with which the performance of the model will be much better. We also learned the model selection evaluation so as to better gauge the model's predicting power and select the best model. Last but not least, we covered feature design and feature selection.

### 2.1 Scalable ML Models

In this semester, we've gone through several fundamental learning algorithms, as well as objective functions and models. Supervised learning problems are those with input features, such as linear regression and classification problems. Linear regression is used when the target variable to be predicted is continuous; when the prediction are discrete values such as predicting positive or negative of a certain disease, it is a classification problem. To minimize the cost functions, Least Mean Square algorithm, matrix derivatives or maximum likelihood estimate can be applied to derive the parameters for models[3]. Logistic regression models assumes there is one smooth linear decision boundary and it is a powerful statistical way of modeling when the dependent variable is a binomial value. It is also the most prevalent algorithm for solving the scale problem in industry given its simplicity, efficiency and robustness. Naive Bayes is a family of algorithms for training such as Gaussian naive Bayes, Bernoulli, or multinomial naive Bayes, assuming that all values of variables are independent of other features[15, 4]. Modeling with Naive Bayes can be simple and efficient

while the conditions can be the constraints.

Supporting Vector Machine (SVM) relies on boundary cases to build the separating curves and obtain the optimal margin classifier. SVM can handle large feature spaces and the boundary cases also enable it to handle missing data such in text classification[17, 16]. We've also discussed about decision tree, or regression tree, which is derived by recursively partition the data space and fitting prediction model within each partition[18, 10, 11, 1]. It is more intuitive and easy to interpret given the tree structure. While the complexity of the algorithm is it's main setback. In order to improve the classification rate, we can also use random forest, which applies a number of decision trees[12].

Aside from these basic modeling methods, we are also going to try ensemble modeling such as AdaBoost, Deep Neural Network and Markov Chain Monte Carlo (MCMC). AdaBoost is a meta-algorithm which is used in conjunction with other learning algorithms to improve performances[29]. The training process is efficient since it picks features which improve the predictive power of the model based on the weight and intermediate result for each round. Therefore, it reduces dimensionality and improves the efficiency of learning as less relevant features are not being evaluated. Deep Neural Network(DNN) is also a scalable model and promising to try in practice. DNN is good with modeling complex non-linear relationships[30, 31]. Also, since it has multiple hidden layers of units between the input and output layers, it enables the composition of features from lower layers, making it possible to model complex data with fewer units.

## 2.2 Efficient Learning Algorithms

Convex optimization [gradient based algorithm, Newton] Parallel. [stochastic] gradient descent/ascent

## 2.3 Model Selection Evaluation

Train, CV, Test 2. Eval metrics [MSE, RMSE, AbsError/L1, ROC, F1 metrics, etc.] 3. Regularization

## 2.4 Feature Design/Selection

1. PCA, SVD, Maximum Mutual Information [selection] 2. [design]

# 3 Experiments with Scalable Machine Learning

After laying a theoretical foundation, I am going to design and experiment large-scale modeling techniques on the dataset from KDD Cup 2012 track 2, which is on advertisement ranking. Clicking ads prediction is of great value to research given the enormous revenue it generates in industry. Predicting click through rate (CTR) is central to multi-billion dollar business in Internet marketing. Contextual advertising, display advertising and search advertising all depend heavily on an efficient learning model that can accurately predict the click through rate[5, 23, 24, 19, 13, 14]. Predicting click through rate is not a brand new topic. Yet new methodologies and approaches to tackle it as well as improve the accuracy confer tremendous vitality to this subject. This thesis will focus on applying large-scale learning techniques and predicting the CTR of ads in a web search engine.

## 3.1 Problem Definition/Description

Search advertising has been one of the major revenue sources of the Internet industry for years. A key technology behind search advertising is to predict the click-through rate (pCTR) of ads, as the economic model behind search advertising requires pCTR values to rank ads and to price clicks. In this task, given the training instances derived from session logs of the Tencent proprietary search engine, soso.com, participants are expected to accurately predict the pCTR of ads in the testing instances.

## 3.2 Data

The dataset is provided by Tencent and from the competition in track 2 of KDD Cup 2012. It includes training data, testing data and other additional supporting datasets. Training data contains149, 639,105 instances while the testing data 20,297,594 instances. They both capture the data between user and search engine, including some shared features such as UserID, DisplayUrl, KeywordID, QueryID, AdID, AdvertiserID, DescriptionID, Position and Depth. In addition, the training data also has impression and

Table 1: Data File Description

| **File** | **Size** | **Records** |
|---|---|---|
| training.txt | 9.9GB | 149,639,105 |
| test.txt | 1.3GB | 20,297,594 |
| KDD_Track2_solution.csv | 244MB | 20,297,595 |
| descriptionid_tokensid.txt | 268MB | 3,171,830 |
| purchasedkeywordid_tokensid.txt | 26MB | 1,249,785 |
| queryid_tokensid.txt | 704MB | 26,243,606 |
| titleid_tokensid.txt | 171MB | 4,051,441 |
| serid_profile.txt | 283MB | 23,669,283 |

click whereas the testing data does not. For each instance, the meaning is that in a case, the user is impressed with the ad for certain number of impressions and clicked the ad number of click times. The supporting data contains token lists of query, description and keyword.

Table 2: Data Field Description

| **Field Name** | **Field Description** |
|---|---|
| UserID | id for user |
| AdID | certain ad users impress or click |
| AdvertiserID | advertiser for ad, a property of ad |
| Depth | number of ads impressed in a session |
| Position | the order of an ad in the impression list |
| KeywordID | the keyword of ad, a property of ad<br>it is the key of keyword token file |
| QueryID | id of the query<br>zero-based integer value, key of query token file |
| DisplayURL | a property of the ads (hashed for anonymity) |
| TitleID | a property of ads, key in title token file |
| DescriptionID | a property of ads, key in description token file |
| Click | number of times the user clicked the ad |
| Impression | number of search sessions in which<br>the ad was impressed by the user |

The training data contains impression and click, where impression is the number of times the ad was impressed by the user, and click is the

number of times the impression transferred into an actual click. For every impression, the user may click or not click the ad, which can be viewed as a binary classification. In the original dataset, the click and impression are not binary-based values. Therefore a modification was made to accommodate binary classification. Each data instance was split into two with equivalent embodiment of the original CTR. For example, an instance with 4 clicks and 10 impressions will be split into 1 click with weight 4 and 0 click with weight 6. Regarding the features in the original dataset, modifications also made to have better prediction model. Each feature is represented by the corresponding value such as UserID, QueryID etc., and based these features new CTR features can be generated. For each categorical feature, we first group them and get the average click through rate for each category, then put the value back correspondingly to the original dataset as an additional column of feature.

## 3.3 Evaluation Criteria

The solution will be validated by area under curve (AUC) of the receiver operating characteristic (ROC) curve. ROC curve depicts the classification performance in a two dimensional graph with false positive (FPR) rate and true positive (TPR) rate. Calculating the area under the ROC curve gives us the probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance[7]. Since ROC is in a unit space, AUC is a value between 0 and 1. In the coordinate, the best prediction method will yield a prediction on the upper left, (0, 1), which means that there is no false positive. This measurement is well known for ranking performance and classification performance. Though arguments exist on the coherency of AUC in measuring aggregated classifier performance, the proposed alternative was proved to be already available in the form of AUC[8, 9].

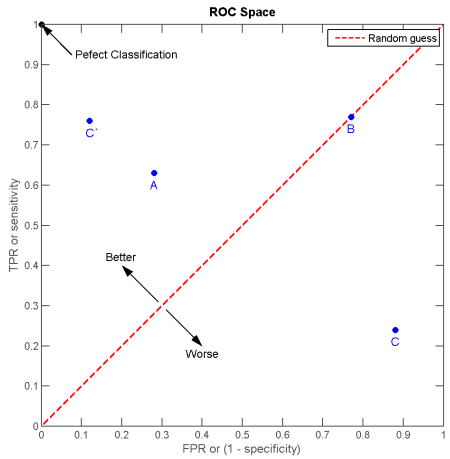


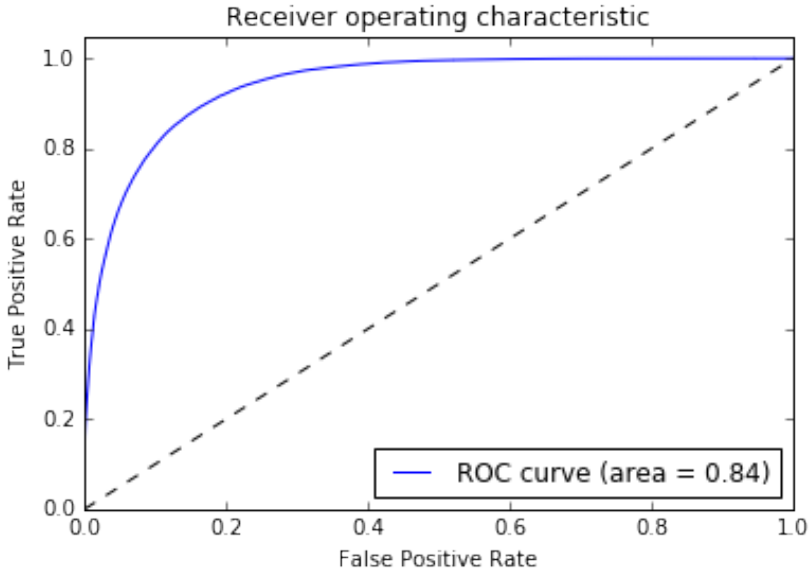

Figure. ROC Space (Wikipedia)

## 3.4 Preliminary Results

Progress has been made on both data pre-processing and some initial modeling. The original dataset contains data on user and search engine, while their values can not be directly used in models. Therefore new CTR features were generated based on the original features. For the dependent variable, which is click, changes were made as well. The raw data contains click value ranging from 0 up to thousands, and the corresponding impression data is an integer larger than click. However, I would like to treat the clicking event as a binary value, either click or not click. Therefore, I made a change to the data on the impression and click. Based on the impression, if it is larger than 1, then the data instance will be split based on how many clicks it has. If the number of clicks is equal to impression, then the click will be updated to 1. If the number of clicks is 0 then no modification is needed. Otherwise, the instance will be split into to two instances. The clicks will be 1 and 0 respectively and impressions for 1 will be the original number of clicks, while impression for 0 will be the original number of impressions minus the original number of clicks. Other than the given features in the original data, new features will be generated from the token files as well. For features such as query, title and description, the token strings will be stemmed using Snowball stemmer. What's more, term frequency-inverse document frequency (tf-idf) will be used for user's query and ads' description[21].

In addition to feature modification, the original data was downsized into a 5 million instances sample in order to process faster locally. The original data contains 150 million data instances with 10gb raw train data as well as token data, user profiles data. Downsizing the data into 5 million instance sample is a valid approach for this stage of proof of concept. On one side, it is faster and more efficient for model selection and feature selection where datasets need to be run multiple times so as to get the best score for features and models. On the other side, 5 million is a sufficient data sample size for initial processing of data and catch the gist for different sets of features and models.

Regarding trying different models, logistic regression model, based on the theoretical view, is efficient for large scale learning and good for its robustness to overfitting. I first tried this model on all the features I have for now. With 5 million instances, I split the data into two parts, training and testing, 70% and 30% respectively for cross validation. Here is the preliminary result after running on my sampled data.




The AUC score is 0.84 which is much higher than the given benchmarks. However, due to the fact that the datasets I adopted are sampled from the original data, the variance is higher so this score might be a little bit higher than fitting the model on the full data, which contains a test file with more than 20 million instances.

### 3.5 Future Work

In next semester, I am planning to learn more advanced scalable machine learning algorithms, models and techniques. In the meantime, this project will be developed into a fully-fledged thesis paper. In order to achieve that, I need to first keep taking this thesis course and study more on the related topics to hone skills in this field. Besides, here is a more detailed plan for my thesis:

First, new features are needed to be explored from the original data. Stemming the tokens, I can then pick valuable information and construct new ads related features from description and keywords, as well as user related feature from query tokens.

Second, model selection needs to be updated given that the new features popping up. Also, with more thorough study on some advanced models, I can try ensemble modeling or feature engineered features such as AdaBoost, Random Forest, SVM, Neural Network etc.

Third, after finding the model with top AUC scores on my sample data, I will need to run the models on the full data, which contains 149,639,105 instances. Due to the size of the datasets and the constraints of my local machine, it is hard to run models locally. So, additional tasks required to mount the data to a Cloud on Amazon Web Service, Google Cloud or Azure.

## References


1. Breiman, L., J. Friedman, and R. Olshen (1984). Classification and Regression trees. Wadsworth. 555 - 597
2. Chan, Philip K., and Salvatore J. Stolfo. "Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection." KDD. Vol. 1998.
3. Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining, Linear Regression Analysis, page 67-119, Edition 5, 2012
4. Drugowitsch, J. (2008). Bayesian Linear Regression. Technical report, U.Rochester.
5. D. Agarwal, B.-C. Chen, and P. Elango. Spatio-temporal models for estimating click-through rate. In Proceedings of the 18th international conference on World wide web, pages 21-30. ACM, 2009

**6.** Duchi, J., E. Hazan, and Y. Singer (2010), Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Retrieved from http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf

**7.** Fawcett, Tom (2006); An introduction to ROC analysis, Pattern Recognition Letters, 27, 861-874. Retrieved from https://ccrma.stanford.edu/workshops/mir2009/references/ROCintro.pdf

**8.** Hand, D.J. Measuring classifier performance: a coherent alternative to the area under the ROC curve. Machine learning, 77(1): 103-123, 2009. ISSN 0885-6125.

**9.** Hanley, James A.; McNeil, Barbara J. (1983). "A method of comparing the areas under receiver operating characteristic curves derived from the same cases". Radiology 148 (3): 839-843.

**10.** Kim H, Loh WY. Classification trees with bivariate linear discriminant node models. J Computer Graphical Stat 2003, 12:512-530.

**11.** Kim H, Loh WY, Shih YS, Chaudhuri P. Visu- alizable and interpretable regression models with good prediction power. IIE Trans 2007, 39:565-579.

**12.** Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.

**13.** M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th international conference on World Wide Web, pages 521-530. ACM, 2007.

**14.** M. Richardson, E. Dominowska, and R. Ragno. Predicting clicks: estimating the click-through rate for new ads. World Wide Web Conference Series, 2007

**15.** Rennie, J.; Shih, L.; Teevan, J.; Karger, D. (2003). Tackling the poor assumptions of Naive Bayes classifiers

**16.** Stuart Andrews, Ioannis Tsochantaridis and Thomas Hofmann: Support Vector Machines for Multiple-Instance Learning. Retrieved from https://www.robots.ox.ac.uk/ vgg/rg/papers/andrews_etal_NIPS02.pdf

**17.** Thorsten Joachims: Text Categorization with Support Vector Machines: Learning with Many Relevant Features, http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf

**18.** Wei-Yin Loh: Classification and regression trees. Retrieved from http://www.stat.wisc.edu/ loh/treeprogs/guide/wires11.pdf

**19.** O. Chapelle. Click modeling for display advertising. In AdML: 2012 ICML Workshop on Online Advertising, 2012.

**20.** Phil Simon (March 18, 2013). Too Big to Ignore: The Business Case for Big Data. Wiley. p. 89

**21.** Ramos, Juan. "Using tf-idf to determine word relevance in document queries." Proceedings of the first instructional conference on machine learning. 2003.

**22.** Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.

**23.** T. Graepel, J. Q. Candela, T. Borchert, and R. Herbrich. Web-scale Bayesian click-through rate prediction for sponsored search advertising in microsofts bing search engine. 2010

**24.** Z. A. Zhu, W. Chen, T. Minka, C. Zhu, and Z. Chen. A Novel Click Model and its Applications to Online Advertising. http://research.microsoft.com/pubs/119092/WSDM2010.pdf

**25.** Article retrieved from http://www.slideshare.net/kmstechnology/big-data-overview-2013-2014

**26.** Article retrieved from http://cloudtweaks.com/2015/03/surprising-facts-and-stats-about-the-big-data-industry/

**27.** Article retrieved from http://blogs.wsj.com/cio/2013/09/11/industrial-strength-analytics-with-machine-learning/

**28.** Article retrieved from https://www.marketingsherpa.com/article/chart/personalized-product-recommendations

**29.** Yoav Freund , Robert E. Schapire: A Short Introduction to Boosting (1999),
retrieved from http://www.site.uottawa.ca/ stan/csi5387/boost-tut-ppr.pdf

**30.** Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural Networks 61 (2015): 85-117.

**31.** Bengio, Yoshua (2009). "Learning Deep Architectures for AI". Foundations and Trends in Machine Learning 2 (1): 1-127.
Retrieved from https://www.iro.umontreal.ca/ lisa/pointeurs/TR1312.pdf