A Filter-based Framework with Support Vector Machine on CIFAR-10 Task

Wei Jiang
weijiang@cmu.edu

ABSTRACT

This is the final project of Introduction to Machine Learning (10-601, Fall 2014) in which over 100 teams compete on the task of classifying images from a subsample of CIFAR-10 [1]. In the project, an empirical study is conducted in order to investigate the performance of different classifiers with feature selection method and Principle Component Analysis (PCA). In addition, a data filtering method is proposed to clean the outliers of the given training data. By using Histogram of Oriented Gradients (HOG) features, the final filter-based framework can obtain a testing accuracy of 60%, which is 8% higher than the baseline implementation.

Keywords

Data filtering, SVM, PCA, Feature selection.

1.INTRODUCTION

As automatic recognition of things in images becomes increasingly eye-catching, various methods of computer vision are proposed. Especially, deep learning is currently one of the most popular machine learning techniques in this domain for its ability of modeling high-level abstractions of image data [2]. Computer vision features such as Scale-invariant Feature Transform (SIFT) , HOG, Speeded Up Robust Features (SURF), etc. are also significant for high performing image recognition systems. Compared with raw pixel data, these kinds of features are able to capture edges, lines, orientations and other important properties of an image [3]. Since the main focus of this project is not computer vision engineering, only HOG features are adopted to evaluate the performance of Bagged Trees, Neural Network, and Support Vector Machine. Along with these algorithms, techniques such as Principle Component Analysis (PCA) and feature selection are also tested. In order to evaluate the effectiveness and efficiency of these algorithms, the parameters of each are fully tuned before the performance are compared.

Another major work on this project is a data filtering pipeline that is proposed to eliminate outliers in training data. The idea, similar to k-Nearest Neighbors algorithm, is to be discussed with detail in section 3. With a careful grid search on different parameters of the filter setting, the predicting performance overwhelms the feature selection methods when the feature space is small. On the other hand, the feature selection method dominates when the feature space is enlarged due to the overfitting problem with this filter. An explanation of this phenomenon is discussed in section 4.5.

2.BACKGROUND

This section describes the dataset and features associated in the project. Related algorithms such as Bagged Trees, Neural Network (NN), Support Vector Machine (SVM), PCA, and feature selection with Information Gain are also discussed. In addition, a detailed summary of the comparative study in milestone one is made.

2.1.Dataset

CIFAR-10 is a labeled subset of the *80 million tiny images* [1]. The training data is a random subset of CIFAR-10 which contains 4000 labeled color images (size of 32*32*3) balanced in 10 different classes, while the test set is a random subset of 15000 images without disclosure of their labels. After the classification model is well trained with appropriate parameters, it will be used for testing set in order to generate predicting labels that can be evaluated with Kaggle. Below is a random sample from the CIFAR-10 with 10 random images from each class [1]:

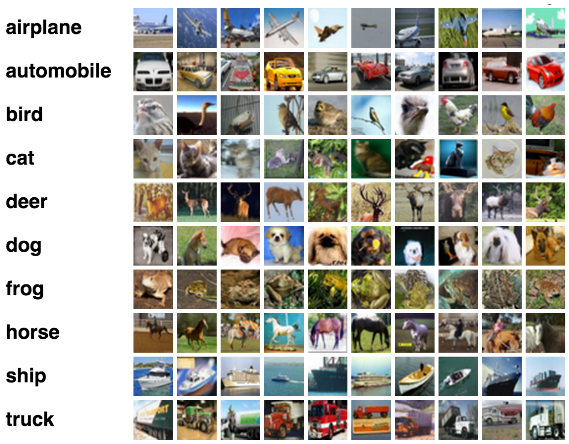

Figure 1. a sample of images in 10 different classes

2.2.HOG feature

Visual features play an important role in computer vision [4]. Although it is not encouraged to do visual feature engineering in this project, a proper feature such as HOG is found more applicable than raw pixels to demonstrate the functionalities of different machine learning techniques that we learned from the class. Compared with SIFT which captures the properties at key points, HOG describes the shapes of a given region in a broader scope [5,6]. Since images in CIFAR-10 are in small size with distinctive shapes of each classes, HOG is believed be a more appropriate feature for this project.

2.3.Related machine learning algorithms

2.3.1. SVM

In milestone one, a comparative study of three different typesetting of algorithms is conducted. The first algorithm is SVM which recently gained popularities in computer vision and many other domains [7, 8]. SVM is a discriminative training process of linear classifier by maximizing the margin hyperplane of classification [9]:

$$f(\mathbf{x}) = \sum_{i=1}^{L} \alpha_i t_i K(\mathbf{x}, \mathbf{x}_i) + d, \tag{1}$$

Since the CFAR-10 contains 10 different classes, I use a multi-label tool with LibSVM in Matlab. Using kernel functions, SVM can often find the hyperplane that linearly distinguish two class which was originally not linearly separable. Based on the test over different kernel methods, the Radial Basis Function kernel (RBF) shows its dominance over others in this project.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2),\ \gamma > 0 \tag{2}$$

Here γ is the kernel parameter. The optimal value of γ is chosen by a grid search with cross validation method.

2.3.2. Neural Network

The second classifier is Neural Network which can approximate non-linear function with multiple hidden layers [10]. Neural network have been proven to be powerful in many domain such as digits classification, forecasting and modeling [11]. Typically, a neural network can be composed of large number of interconnected processing elements that are embedded in the hidden layer in order to learn high level features from original input layer. An example of the structure of a neural network is shown below.

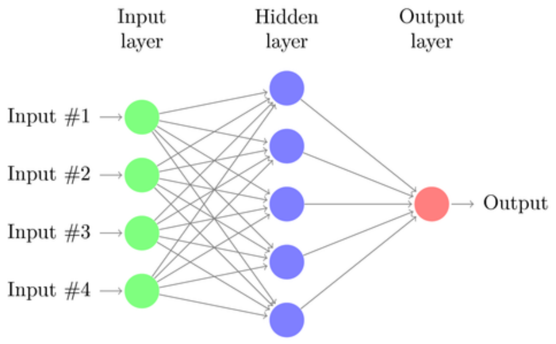

Figure 2. Representation of a one-layer neural network

In this project, a multilayer neural network is constructed with the Neural Network Toolbox. In despite of its advantages, neural network cannot provide a explicit learning process of its hidden layers. In order to learning high quality hidden features, it is often necessary to increase the number of hidden layers. In addition, spending more time to construct a neural network with more numbers of hidden layer may not give expected result.

2.3.3. Bagged trees

The third one is a type of ensemble method which constructs a set of classifiers with a voting process. Common ensemble methods include bagging, bucket of models, stacking and boosting. In this project, a bagged trees classifier is adopted by using the Statistics Toolbox. During the training process, the original train set is randomly sampled to be many bootstrap replicates [12]. After each tree classifier is trained without pruning, the result is to be voted. With an appropriate size of resamples, bagged trees can lead to an acceptable prediction accuracy competing with SVM. Due to its low efficiency, this method is not adopted in milestone two.

2.3.4. Principal Component Analysis

PCA is mainly a feature de-correlation process which can help to reduce dimensionality without losing too much information [13]. By projecting the data to a new coordinate system, the variances of each dimension are ranked with the order of coordinate. In general, top principle components can often represent reserve most of the properties the original data has. By selecting a proper number of principle components, the machine learning pipeline can ofter run faster.

2.3.5. Feature selection

In machine learning, feature selection plays an important role in constructing a well-performed model. By selecting a subset of features which contain the most relevant information, we can build a model without confusing it with other redundant or irrelevant features. Although other typical feature selection methods such as Chi squared, OneR, etc. are also adopted and evaluated, only information gain is implemented in this project.

2.4.Implementation of baseline in milestone one

The major focus of milestone one is to investigate both the efficiency and effectiveness of three different machine learning methods. The input data is the training set which contains 4000 instances. With feature extraction of HOG, the feature dimension drops to 324 with the default parameter setting of the Matlab function “extractHOGFeatures”.

For SVM classifier, the training function is adopted from the LIBSVM package which provides functionalities such as multi-class classification and cross validation for model selection. By using the command “svmtrain(y, X, sprintf('-s 0 -t %f -g %f’, gamma))”, a trained model will be contructed. In the options, ‘-s 0’ indicates a multi-class classification; ‘-t 2’ indicates the RBF kernel; and ‘-g gamma’ indicates the value of gamma in the RBF kernel. Although linear kernel and polynomial kernel are also adopted, RBF kernel is found to be the best. With a careful cross validation experiment, the optimal gamma value for RBF kernel. Finally, the command “svmpredict(testLabel, testData, model)” is used to evaluate the model learning by the best gamma value of RBF kernel.

For neural network, Matlab provides a usable Neural Network Toolbox. The command “net = patternnet(k)” is used to build the architecture of a neural network with k hidden layers. The network ‘net’ is then trained by “train(net, X, y, ‘UseParallel’, 'yes', 'UseGPU', 'yes')". Options of 'UseParallel' and 'UseGPU' can help with the efficiency of the training process. After the network 'net' is fully trained, it can be used to predict the test set by using "testY = net(testX)". In this application, number of hidden layers k is the only but significant parameter setting. Although a larger value of k is believed to generate a more complex network with better performance, an oversized k often makes the network more expensive to compute and sometimes may degrade the performance. From the experiment, a 200-layer neural network quite fit the data.

For bagged trees, the bagging algorithm provided by the Statistics Toolbox from Matlab is used. After the HOG feature is extracted, the data is randomly partitioned into a training set and a testing set. By using "bag = TreeBagger(numTrees, Xtrain, Ytrain)", a tree bagger is learning with choice of numTrees. The predicting label can be obtained by "[Ypred, ~] = bag.predict(Xtest)" and evaluated by comparing with the testing labels. Similar as neural network, a larger number of trees in the bagger may improve the predicting performance. However, an oversized choice of number also cost more time to train the bagger and may degrade the performance.

All of three algorithms implemented in milestone one provide acceptable testing accuracy over 40% with the 5000 testing set in Kaggle. Among these three classifiers, neural network is much faster than the other two methods with parallel features, but SVM with RBF kernel leads to the best performance. The detailed comparison of both efficiency and effectiveness is to be discussed in the result section.

3. METHODS: An empirical study and a proposed data filtering method

The main task in milestone two is to improve the performance from the baseline. PCA and common feature selection methods such as information gain are adopted with both raw pixel features and HOG features. More importantly, a data filtering method is proposed which overwhelms feature selection according to the experiments. The main motivation of developing the data filter lies in that computer visual feature engineering is not encouraged in this project. The experimental result is to be discussed in section 4, while this section focuses on the description of the methods.

3.1.Use PCA to reduce the dimensionality

In this project, the original raw pixel feature is in a high dimension space. Both the training and testing with original feature take long time. By applying PCA to the original 3072 dimension feature, the experiment shows that the first 300 principle components give the similar prediction accuracy but take much less time in all of the three classifiers that mentioned.

Since PCA is a de-correlation process [14], one of the limitation of it is that dimension reduction can only be effective if the original features are correlated. After HOG feature is extracted with cell size of 8 and block size of 2, the feature space turns out to be a much lower dimension of 324. The experiment shows that the implementation of PCA only degrades the performance of the pipelines despite of numbers of principle components chosen. Consequently, the final pipeline does not include PCA.

3.2.Use feature selection with information gain

In general, information gain is a measure of information that inherited by a feature [15]. Because information gain is biased in favor of features with large values, a normalization process is often needed prior to feature selection [16]. In the experiments, I use Weka 3.6.11 to select attributes and generate a text file for further evaluation in Matlab. By selecting different numbers of features after feature selection, three different classifier are evaluated accordingly. The experiment shows that feature selection method cannot improve the performance with default setting of HOG features.

3.3.Proposed pipeline of data filter

In order to advance the performance of the implementation to a higher level, the main focus of the project is shifted from feature engineering to data filtering. Since the main feature used in this experiment is HOG which describes the shape of a given area, shapes in given images might be a distinctive factor. However, some objects in the images are in various orientation or perspective which make them hard to be distinguished even with human eyes. Consequently, if there exists a filter that can eliminates instances whose representation are not distinctive with other types or not coherent with images in the same classes, a more powerful classifier can be learned from a "cleaner" training dataset.

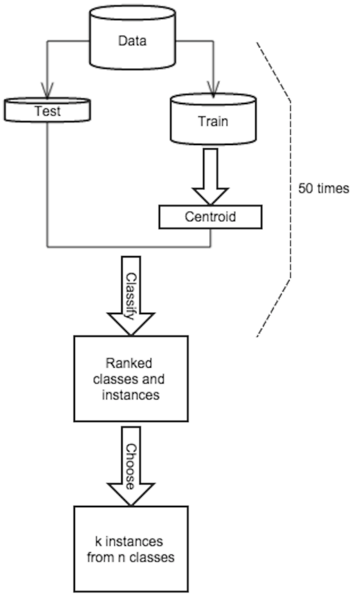

Figure 3. Proposed pipeline of data filter

Figure 3 shows the pipeline proposed in this project. HOG feature was firstly extracted from the original labeled data prior to the pipeline. Then the data is randomly partitioned into a training set and a testing set. After the centroids of each class in the training set are found, a k-nearest neighbor is adopted to classify the testing set based on the euclidean distance to each centroid. By running this process multiple time, a confusion matrix with consistency can be constructed. The pipeline then ranks the 10 classes and instances based on the error rate associated with each of them. By choosing k instances among the top n classes based on the value of error rate, the indexes of a set of outliers are selected by the filter.

The Matlab function of the usage is "[X, y, removdedIdx, accEachClass, idx_rank] = dataFilter(Data, Labels, n, k, partitionRate)". By feeding the input data, labels, number of classes and instances, and portion rate, it will return the indexes of the filtered data. The choice of number of instances should not be too large since removing a large number of training instances will only degrade the perform of the pipeline. In the implementation of this filter, a grid search method of the parameters is done in order to find the most appropriate combination of parameters. With a set of thorough grid searching studies with SVM (RBF kernel), the optimal value for the filter settings are "0.2, 2, 25". According he the experiment, this proposed pipeline can improve the testing accuracy to 60% using the HOG feature with a dimension of 324. The experimental outcomes are discussed in the following section.

4.Experimental result

The experimental result of both milestone one and two are discussed in this section. In milestone one, the performance of three classifiers are evaluated in section 4.1 ~ 4.3. In section 4.4, an analysis of PCA is conducted over the raw pixel features. Finally in section 4.5, the proposed data filtering pipeline is also evaluated by a comparison with feature selection.

4.1.Compare kernel functions of SVM

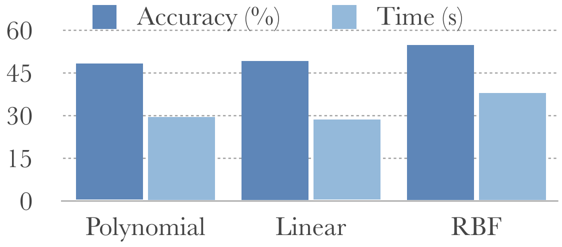

Figure 4. SVM with different kernel functions

Figure 4 shows that with a well-design cross validation experiment, RBF kernel performs the better than polynomial and linear kernel. It is worthy to mention that parameters of each of them are fully optimized. Although RBF takes longer time per fold, the efficiency can still acceptable.

4.2.Choice of hidden layers of neural network

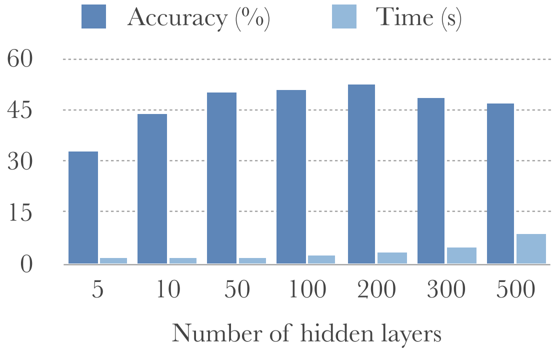

Figure 5. Neural network with choices of hidden layers

Although a large amount of hidden layers typically provides better performance for neural network, the experimental result shows that a neural network with 200 hidden layers performs better than other choices. On the other hand, even with 800 hidden layers, the neural network takes less than 15s to train on the data.

4.3.Choice of number of trees of bagged trees

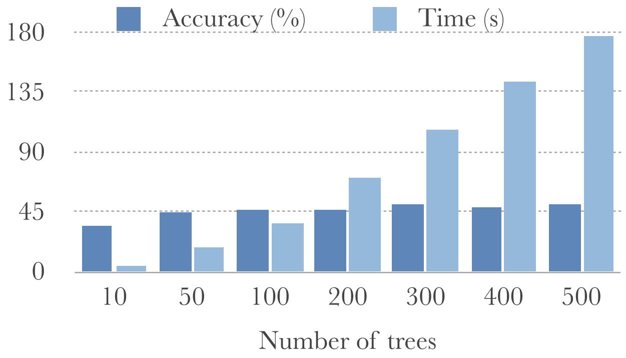

Figure 6. Bagged trees with different choice of trees

The above result shows the performance of a simple tree bagger model on the on the 324 HOG feature. Compared with SVM and neural network, the prediction accuracy of bagged tress is generally lower. Furthermore, the classifier takes more than a minutes to train the bagger when number of trees reach to 200.

4.4.PCA of raw pixels

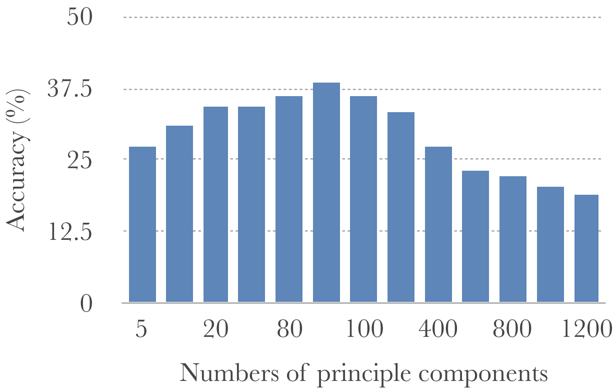

Figure 7. Evaluation of PCA

PCA is often used for dimension reduction. With the large feature dimension space of raw pixel, the top 90 principal components can dramatically decrease the time of training the classifier with a testing accuracy of 38.3% with SVM. Unlike HOG feature which is extracted from images in gray scale, pixel features reserve the information of colors. For example, ships often come with ocean which is in blue, while frogs often show brown skin on a green background. Although PCA alone cannot lead a better performance than HOG feature, combining the top principle components with HOG is proved to be effective in the proposed framework.

4.5.Data filter v.s. feature selection with different numbers of HOG features

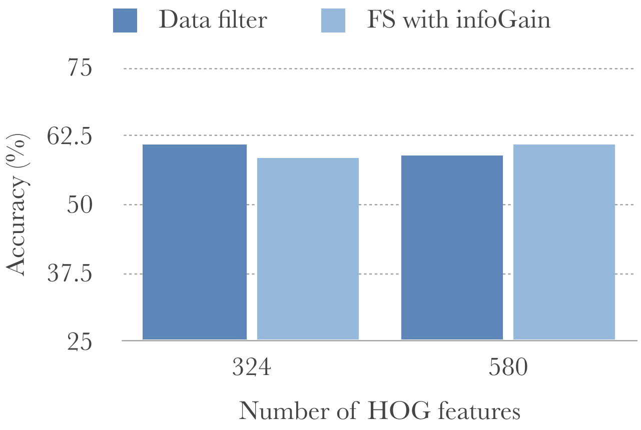

Figure 8. Data filter v.s. feature selection

Since adding new feature to a machine learning framework is a typical way to improve the performance, an additional set of 256 HOG features is introduced by setting the cell size and block size both at 4. However, figure 8 shows that there is no significant improvement with the proposed filter when using the combined HOG features with a large size. With this larger feature size, feature selection with information gain can actually improve the predicting performance by ranking the features. This phenomenon can be explained by the the idea of bias and variance. Since the additional features are also HOG, some of them may be similar to the previous 324 features with a different setting. The addition with this redundant features may overfit the classification process of the filter. On the other hand, feature selection method is able to capture the property of redundancy in the combined features. Considering the fact that the size of given training data is only 4000 which is comparatively small, the proposed filter should work significantly better if more training data are used

5.CONCLUSION

In this project, SVM is chosen for the final experiment based on its performance on the baseline implementation. In addition, studies of PCA and feature selection are conducted. Although feature selection does not perform well with a small number of HOG features, it leads a slightly better accuracy over the proposed filter when the HOG feature is in a larger dimension. The purpose of the proposed data filter is to eliminate outliers in the given training data. By deleting a proper numbers of instances for the training, the testing accuracy increase by 8% in average with a small number of HOG features. Although, a continuous study with additional HOG features also indicates a high variance when using the data filter, training data of real application is often significantly larger than 4000 and outliers are often inevitable. The proposed data filter should be prominently more effective. For future works, more efforts can be put on measuring the distance between class centroids and instances. A well-engineered method of choosing numbers of outliers can be also explored.

6.ACKNOWLEDGMENTS

Thanks to the support from Dr. Ziv Bar-Joseph, Jin Sun and Yipei Wang.

7.REFERENCE

1. Alex, K. 2009. Learning multiple layers of features from tiny images
2. Honglak, L., Roger, G., Rajesh, R., and Andrew Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. ICML '09, 609-616.
3. Andreal, V., and Brian, F. 2010. Vlfeat: an open and portable library of computer vision algorithms. ACM, New York, NY 2010, 1469-1472
4. Kin, C. Y., and Roberto, C. 1997. Feature-based human face detection. Image vision. Vol. 15, 713-735
5. Dorin, C. and Peter, M. 1997 IEEE Computer Society Conference. DOI= http://dx.doi.org/10.1109/CVPR.1997.609410
6. Benjamin, C., David, B., Philip, M. and Jitendra, M. 2008. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies*. DOI= http://dx.doi.org/10.1016/S0968-090X(98)00019-9
7. Campbell, W. and Sturim, D. 2010. SVM based speaker verification using a GMM supervised kernel. MIT Lincoln Laboratory
8. Keerthi, S., Shevade, K., and Murthy, K., 2001. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation 2001. Vol. 13.657-687*
9. Leslie, C., Eleazar, E., and Stafford, W., 2001. The spectrum kernel: A string kernel for SVM protein classification. *Pacific Symposium on Biocomputing. 566-575*
10. Rowley, H., 1998. Pattern Analysis and Machine Intelligence, *IEEE Transactions, Vol 20. 23-38*
11. *Kurt, H., 1989.* Multilayer feedforward networks are universal approximators. *Neural Network. Vol. 2. Issue 5. 359-366*
12. Tomas, G., 2000. Ensemble methods in machine learning. *Lecture notes in computer science. Vol.1857. pp 1-15.*
13. Moore, B., 2009. Principal component analysis in linear systems: Controllability, observability, and model reduction. *Automatic Control, IEEE Transactions on Vol. 26. Issue. 1.* DOI: http://dx.doi.org/10.1109/TAC.1981.1102568
14. Pierre, B., and Kurt, H., 2001. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks. Vol. 2, Issue 1, 53-58.* DOI: http://dx.doi.org/10.1016/0893-6080(89)90014-2
15. Isabelle, G., and Andre, E., 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research. Vol. 3. 1157-1182*

8.Appendix

This project is independently conducted by Wei Jiang. I have implemented a great number of machine learning techniques learning in 10-601. Not only I have a better under standing of algorithms such as SVM, neural network, ensemble method, and feature extraction, but also I experienced a sufficient exercise of coding in Matlab. Especially, when I could not find a way to improve the performance of the framework in milestone two with feature engineering method, I started to change my focus on given data. By implementing the proposed filter, part of the outliers in the training data were eliminated. As a result, the final framework performed well with only HOG features.

Although 10601 is an introductory level course of machine learning, hands-on projects are definitely significant part of our study. However, there are still some parts of the project that can be improved. Firstly, the project is a task of computer vision which highly relies on computer vision features. If the future project can be a broad range of topics, it will be more appropriate. In addition, the time that we have for this project is not enough, especially for students who cannot find a partner. Future projects may give students more time.