CIFAR-10 Image Classification Using K-Mean Clustering and Multinomial Logistic Regression

Graeme Rock
Team Terminator
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
+1 (843) 813-2477
grock@andrew.cmu.edu

Alvin Chou
Team Terminator
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 1513
+1 (678) 770-9212
alvincho@andrew.cmu.edu

ABSTRACT

In image classification, our team explored the concept of using feature selection and K-mean clustering techniques to identify higher-level features based on a subset of CIFAR-10 RGB data. Thereafter, we compared the use of Multinomial Logistic Regression to K-means and the use of MLR to perform classification on the higher-level features to train and test the given CIFAR-10 dataset. Through our exploration and experimentation, we seek to improve on the baseline approach of using MLR on the basic RGB data with a regularization of 1 loss coefficient of $1e^{-2}$ by 5%. However, after using multiple combinations of feature selection and tuning techniques, we were not able to achieve a better result than the baseline approach.

1. INTRODUCTION

For classification methods in general, K-means and Multinomial Logistic Regression are some of the simplest and most effective methods in clustering and classification of categorical classes respectively.

In our application, compared four forms of classification on the CIFAR-10: Multinomial Logistic Regression (MLR), K-means "Bag of Words" classification, MLR on the extracted feature vectors, and MLR on the clusters groupings created by K-means. We extracted filter responses as a preprocessing filter to allow us to extract high-level features. We then use K-means as a second preprocessor in order to reduce the complexity of data to be classified. Then we used "Bag of Words" and Multinomial Logistic Regression method to classify both groups of high-level features. Our motivation is that by having a higher-level features than simple RGB data, this would allow us to build better correlation and more meaningful feature sets in the classification process than RGB. From then on, we look to improve upon this method by having a higher accuracy result than the initial SVM process with RGB features.

Our main motivation for using K-means and Multinomial Regression is our good understanding of how they function and the code used to apply them due to our prior experience with them. They are highly appropriate toward this application, as MLR classifies fairly accurately on its own and K-means has been shown to segment features of images clearly and efficiently. We steered away from many other classifiers due to their unbearably long runtimes, whereas MLR takes less than a minute and K-means can be completed in about half an hour on a large data set.

We chose to observe the accuracy of classification using the feature vectors themselves as well because we predicted MLR could use feature vectors just as well as pure RGB values in pixels to differentiate pictures. Because "Bag of Words" is commonly used as a basic form of computer vision, we used it for verification that we obtained feature vectors and implemented K-means correctly and to compare its results to MLR for validation of one versus the other.

2. BACKGROUND

In our baseline approach, we explore using a MLR and a SVM with no preprocessing of the input CIFAR-10 data set. The MLR/SVM is trained based on the RGB value of 4000 32x32 images in the CIFAR-10 database. Our initial baseline test uses Multinomial Logistic Regression with a regularization parameter of 1. In our initial baseline approach, we achieve a 38% accuracy.

Multinomial Logistic Regression approach expands upon the traditional Logistic Regression method to allow for the classification of more than two classes. Since our CIFAR-10 database contains 10 different classes, including, trucks, automobiles, birds, cats, dogs, deer, airplanes, frogs, horses, and ships, it is simply impractical to run a traditional Logistic Regression. Similar to how Linear Regression separates data using a decision boundary, Multinomial Regression uses multiple decision boundaries to classify more kinds of data. We use a Radial Basis Function Kernel in our MLR, which separates data radially from the origin. For tuning parameters, the regularization term λ is used to prevent over fitting of training data through the use of penalization on the regression. On the other hand the loss function γ specifies the penalty of misclassification and is negatively correlated to the number of misclassifications that occurs.

K-Means is a simple unsupervised learning approach where the algorithm attempts to cluster and group a set of feature points together based on proximity of data points to one another. The algorithm starts with K randomly selected starting points from the data set as the initial cluster centers and assigns all other points in the data set to whichever of the initial points it is closest to using Euclidean distance. Then the algorithm finds the center-most point of the newly created cluster and again assigns points to clusters based on which center is closest. This process continues until the central points no longer change, whereupon the current centers are output as the cluster centers.

"Bag of Words" is a classification method where K-means is performed on the extracted feature vectors of training images, creating a dictionary of "words" that are the K cluster centers. Feature vectors of test data are compared to these centers and assigned cluster center values as well. Histograms are then created for each training and testing image that measures the proportions of clusters to which its feature vectors belong. Each testing histogram is compared to each training histogram and is assigned the label of the image that it is most similar to.

In our experiment, we decided on using an approach where we start out with a baseline method using the default dataset. Then we try to make small incremental improvements by increasing the meaning of our feature set and then by tuning our parameter.

In addition to our established approach, we have also explored alternative methods to be used for classification. In particular, in establishing our baseline test runs, we experimented with using Multi-layered Perceptron, SMO, Random Forest, and Bayesian Networks. In addition, we explore implementing PCA dimension reductions using WEKA 3.7. Given our preliminary testing, however, we find that Multinomial Logistic Regression yields the highest baseline accuracy at 38% while other methods were significantly inferior with the parameters we used. In addition, for Multi-layered Perceptron and PCA dimension reductions, the computational effort was too intensive that we had to abort or implement sampling of features. For Multi-layered Perceptron, each cross-validation fold was taking longer than 2 hours to complete, using a 10-fold cross validation method. we could have reduce the number of hidden layers from (number of attributes + number of instances)/2 to a much smaller amount to accommodate for the computational time, however, we decided that Multinomial Logistic Regression to be our baseline approach.

3. METHODS

3.1 Feature Extraction

Given the initial 4000 training dataset of CIFAR-10 images, our initial approach is to perform feature extraction on our dataset since RGB data is insufficient in terms of information provided. Using the feature extraction algorithm provided by Carnegie Mellon University's 10-601 course in Assignment 6. The feature extraction process takes in the RGB data of an image and output the image with 99 dimensions for each pixel. Given this set of feature points, we used a random sample of 1%, 2%, and 3% of the feature vectors and their feature points for building dictionary and k-means clustering.

3.2 K-Means Clustering

K-means clustering allows us to associate any given set of data within a defined group of data with similar attributes. Our goal is to use the output of our clustering result on the training set as the basis of our Multinomial Logistic Regression. For clustering, we first create a dictionary of cluster centroids using the feature points we have obtained through the feature extraction process specified above. In order to create the cluster centroids, we used a k-means clustering algorithm with 300 clusters and a stopping condition where sum of changes in distance to each respective cluster assignment is less than or equals to 0.

After the baseline dictionary has been created, we then run our initial 4000 training dataset of CIFAR-10 images through our dictionary using the same k-means conditions, creating 4000 32x32 images where each value represents the cluster belonging of its respective pixel. At the same time, we also performed the same task of cluster assignment through k-means using the 15000 testing dataset provided. Given this cluster representation of both the training and testing dataset, we expect our Multinomial Logistic Regression can provide better results than the baseline approach after tuning. For performance measures, the feature extraction and k-means calculation took a total of about two hours to perform, including the initial feature extraction, dictionary creation, and performing k-means on both our training and testing dataset. In addition, the processing speed is highly dependent on the number of samples of feature vectors we decided to use.

3.3 Multinomial Logistic Regression

In our use of Multinomial Logistic Regression, we adapted our base algorithm from Carnegie Mellon University's 10-601 course Final Project Assignment. From our results in the preliminary experiments, we decided to use Radial Basis Function kernel instead of a polynomial kernel to reduce on the computational effort required in our experiment. Given the different feature sets available, we decided to use the following feature sets for each pixel in our experimentations: RGB values, clustered assignments, and 10%, 20%, and 30% of the additionally extracted features. In addition, despite the understanding that excessive features have diminishing return on classification accuracy, we nevertheless tried combinations of the feature sets above, including RGB + additionally extracted features and RGB + clustered assignments. We started our implementation of MLR with single validation on the training set. Since our baseline approach, where $\lambda = 1$ and $\gamma = 1e^{-2}$, seems to overfit our training set and produces poor results on our testing set, we varies the choices of parameters of the MLR, and experimented with different λ values, from 1 to 10 and γ from 0.5 to $5e^{-3}$. In estimating the λ value, we will first try a prior experimental approach where λ value is adjusted according to the results of the last trial. By observing the decrease in accuracy in the training set, we expected a smaller decrease in our testing set, thereby confirming our speculation of the over fitting situation. Next, we attempted to optimize our kernel width regularization λ value by using an approach from Alex Smola [1]. The method derives from finding the 0.1 quantile, 0.9 quantile, and the median of the distance between each data pairs. Then, by taking the inverse of the three values, we can experiment on them with our training dataset and obtain a good approximate of our kernel width λ. By modifying various tuning parameters we look to reduce over fitting on the training set while improving the base model of our classifier. In terms of performance, the time complexity of our algorithm varies depending on whether we use cross-validation or not.

3.4 "Bag of Words"

Using the dictionary found by K-means, we found the cluster to which each of the feature vectors in the testing and training data subsets belonged.

We hoped this might improve upon our baseline result as it is a classifier that is natively used in computer vision. However, a double-edged sword to using "Bag of Words" method is that there are no parameters to tune, other than the K value for K-means and the number or size of feature vectors used.

4. RESULTS

In our preliminary experiments, we performed classification using multiple methods on the RGB feature set to try and find the best classification result. Then, using the method with the best result, we tuned the parameters of the method as well as the choices of features to try and improve upon the baseline we established in the preliminary experiment. For the preliminary experiment, out of choices among Multi-Layered Perceptron, Bayesian Networks, Support Vector Machines, Multinomial Logistic Regression, Random Forest, and Sequential Minimal Optimization, we found that Multinomial Logistic Regression yielded the highest result. In Multi-Layered Perceptron, we chose to reduce the number of features used for sampling to reduce on the time complexity. In addition, in an effort to reduce high dimensionality of our data, we tried to perform PCA dimension reductions on the dataset using WEKA 3.7. However, the time complexity of the process exceeded our computational bandwidth and we decided to proceed using the full features.

In our tuning of the Multinomial Logistic Regression, we were not able to improve upon the baseline we have established in the preliminary experiment. Despite the use of various feature sets and parameters, the baseline algorithm achieved the highest accuracy at 100% on the training set with RGB values and 42% on the testing set using the same features. The parameters used for the baseline MLR includes λ of 1 and γ of 1e^{-2}, and even though we believed that the model created an overfit situation, we were unable to achieve higher results despite augmenting conditions to reduce over fitting.

4.1 Preliminary Experiments

In our preliminary experiments, we are interested in exploring the accuracy of our 4000 CIFAR-10 training dataset on 15000 CIFAR-10 testing dataset by using multiple classifiers and evaluating the results among them. In our experimentation, we decided to observe classifications commonly used in image classification and classification of CIFAR-10 data. As such, our choices of classification includes Multi-Layered Perceptron, Support Vector Machines, Multinomial Logistic Regression, Bayesian Network, and Random Forest.

From our experimentation of various classifiers, we found that dual formed Multi-Layered Perceptron with RBF kernel resulted in the highest testing accuracy at 42% using a low regularization term of $\lambda = 0.1$ and loss coefficient of 1e^{-2}. This result provides a good baseline for us to improve upon as the result is significantly higher than the random assignment approximation of 10% given 10 classes of data. From our observation, the errors deriving from the low accuracy is due to the inability of the classifier in separating classes with similar features and the lack of information in the given feature set of only RGB data.

Table 1. Preliminary Experiment Results

Classifier	**Testing Accuracy**	**Comments**
Random Forest	22.4%	Random sample of 500 instances
SVM	38.3%	Parameters?
Bayesian Net	20.6%	Random sample of 500 instances
Multi-Layered Perceptron	36.7%	Random sample of 10% of attributes
Multinomial Logistic Regression	42.3%	$\lambda = 0.1$, dual form, RBF kernel, $\gamma =$ 1e^{-2}
Multinomial Logistic Regression	9.7%	$\lambda = 0.1$, dual form, poly kernel = 4, $\gamma = 0.5$

4.2 Development and Evaluation Data

From our baseline approach, we look to improve upon the Multinomial Logistic Regression where 42% accuracy was achieved in our preliminary experiment. In our observation of the data, we look to improve upon the baseline following two main approaches: 1) to extract more meaningful features to use in the training and testing model than the initial RGB data. 2) To tune the MLR to reduce over fitting conditions on the training set.

Our first approach to improve the baseline accuracy is to improve on the quality of the features being used in evaluation. Using the feature extraction and k-mean clustering techniques, we were able to create additional feature points and cluster associations for our images. We applied random samples of 10%, 20%, and 30% on our newly extracted 99-dimensional features and performed MLR using the same initial parameters with dual form and RBF kernels. However, we were unable to achieve any testing accuracy above 10% using the approach defined. Even after accounting for tuning of various λ and γ values, our results topped off at 13% using 30 of the extracted features points for each pixel.

Table 2. MLR Using Extracted Features per Pixel*

Per Pixel Feature Size	**λ**	**γ**	**Test Accuracy**
10	0.1	1e^{-2}	12.1%
20	0.1	1e^{-2}	10.4%
20	1	1e^{-2}	10.4%
20	300	1e^{-2}	10.4%
30	0.1	1e^{-2}	12.1%
30	0.5	1e^{-2}	11.8%
30	0.1	1e^{-3}	13.1%

*each image used for training and testing has (32x32xPer Pixel Feature Size) features

From our observations, the use of our extracted feature vectors do not improve upon the initial preliminary baseline. In fact, our extracted feature vectors is similar to the results of random classification. It is a possibility that errors occurred in our use of the feature extraction tool provided in CMU's 10601 course.

In addition, we look to improve upon the baseline by using the extracted features in addition to the original RGB values. Here our expectation is that by adding in additional information regarding each pixel, we can improve upon the classification process.

Table 3. MLR Using RGB plus Addition Features*

Per Pixel Feature Size	**λ**	**γ**	**Test Accuracy**
3 (RGB) + 10 Extracted Features	1	1e^{-2}	25.4%
3(RGB) + 10 Extracted Features	10	1e^{-2}	25.4%
3(RGB) + 30 Extracted Features	0.1	1e^{-2}	14.6%
3(RGB) + 30 Extracted Features	50	1e^{-2}	12.7%
3(RGB) + 30 Extracted Features	0.2	1e^{-2}	14.6%

3(RGB) + 30 Extracted Features	0.2	0.5	9.7%

*each image used for training and testing has (32x32xPer Pixel Feature Size) features

From our observations, adding in RGB features in addition to our extracted features did improve upon our initial results of using only extracted features. However, this is to be expected since with only RGB features, the results were at the 42% benchmark compared with the results achieved above in Table 2. In essence, when using additionally extracted features through the feature response algorithm, our results worsen as more noise is added to the results.

Table 4. MLR Using Clustered Representation as Features with 300 Possible Clusters

λ	γ	Training Accuracy	Test Accuracy
0.1	1e-2	100%	9.7%
1	1e-2	100%	17.9%
10	1e-2	100%	17.9%

For our next experimental data, we look to use our K-means clustered results of each pixel as our features for MLR classification. We started with using the parameters identical to the baseline parameters (λ = 0.1, dual form, RBF kernels, & γ = 1e^{-2}). Using these data with our clustered features, we found our accuracy was dramatically lower at 9.7%. Since we achieved a 100% training accuracy, we suspected that over fitting was a key problem using this approach. As such, we increased the regularization term λ to 1 and 10. In both cases, the train accuracy achieved the same 100%, but the test accuracy were only slightly higher at 17.9%. From here, we decided that our approach to using higher-level classification to achieve better accuracy was inconclusive and continued to tune the parameters using the default RGB dataset.

In our subsequent experimentation, we look to improve upon the baseline model by varying the different parameters in the MLR classification. In this context, we decided fix our use of RBF kernels, but varies the λ and γ values. Given our preliminary results of 43% using λ = 1 and γ = 1e^{-2}, we believed that our model is over fitted as there are very little overfit reduction conditions employed.

Table 5. Optimization of MLR with Various Λ and Γ Values

λ	γ	Train Accuracy	Test Accuracy
0.1	1e^{-2}	100%	43%
0.09	1e^{-2}	100%	42.67%
0.2	1e^{-2}	100%	27.6%
0.467	1e^{-2}	100%	41.9%
0.1	3e^{-2}	99.92%	37.5%
0.2	3e^{-2}	95.35%	35.4%
0.3	1e^{-1}	100%	28.8%
1	0	10.4%	25.6%
0.1	0.5	100%	26.4%
2.1427	1e^{-2}	100%	33%
2.1427	2e^{-2}	69.95%	34.1%
2.1427	3e^{-2}	30.9%	30.5%
2.1427	4e^{-2}	93.25%	27.33%
2.1427	9e^{-2}	99.98%	28.9%
2.1427	1e^{-3}	29.28%	29%
2.1427	5e^{-3}	49.3%	39.1%
2.1427	4e^{-3}	40.5%	37.1%

From the observations, we were not able to achieve a higher testing accuracy despite the tuning of our parameters using multiple methods. By increasing the regularization term to reduce over fitting, from 0.1 to 1, we saw a decrease in testing accuracy, but not training accuracy. As such, our tuning of regularization term did not provide any functional improvements to our model. Possible reasons for this could be that the data we used on the training set did not have significant outliers in terms of data parameterization and that the regularization term we used was therefore not significant enough to have an impact on the final model. Next, we tune the loss function, γ, to allow higher tolerance in misclassification scenarios. By reducing the loss function, we allowed less penalties in misclassification, leading a decrease in training accuracy. This, however, did not have a positive impact on the test accuracy and our test result on this set of tuning varies from 29% for 9e^{-2}to 39.1% for 5e^{-3}. Some reasons for the lack in increase in testing data could be that despite higher allowance in the misclassification of data, our test data contains dataset drastically differs to those of the training set. Even though we reduced the penalty for misclassification, we did not have enough meaningful features without advance feature filtering to produce a quality model for image classification based only on the RGB values of our training dataset.

5. CONCLUSIONS

For this experiment, we compared the results of MLR classification on RGB values versus higher-level features using feature vector extraction and K-means clustering. We also compared with "Bag of Words" classification as a validation of MLR versus a traditional computer vision method.

If we were to continue running this experiment, increasing lambda and decreasing gamma would be our next course of action because it allows higher tolerance in misclassification while penalizing over fitting. We also believe if we had the computer power to process larger sets of data, we could have found more interesting results for classification using K-means.

We also found that the code we wrote to extract the data's feature vectors was slightly erroneous, which may have given incorrect results for our experiment.

6. BIBLIOGRAPHY

[1] Smola, A. Easy Kernel Width Choice. 2009. http://blog.smola.org/post/940859888/easy-kernel-width-choice.

[2] Musavi, M.T., Ahmed, W, Chan, K.H., Faris, and K.B., Hummels, D.M. 1992. On the training of Radial Basis Function Classifiers. University of Maine. http://www.sciencedirect.com/science/article/pii/S0893608005800383

7. APPENDIX

In this experiment, both Graeme and Alvin were involved in identifying the initial baseline approach. During the preliminary experiments, each team member was responsible for testing different classifiers and identify a baseline with reasonably high accuracy to further explore in the subsequent stages. In the next phase of the experiment, Graeme was mostly responsible for adapting and writing code to perform feature extraction, K-means and “Bag of Words” classification while Alvin was responsible for completing the majority of runs to tune MLR parameters.

From this project, we learned the difficulty of tuning the parameters of a simple classifier for balancing training and testing error, as changing them slightly can cause the results to vary wildly.

We also learned the benefits of using MLR for its surprising accuracy and agility in classification. Graeme learned how to perform the “Bag of Words” as a classification method of K-means results other than MLR.