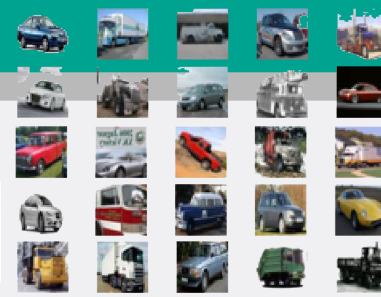
OBJECT CLASSIFICATION FOR AUTONOMOUS VEHICLE NAVIGATION

SJSU SAN JOSÉ STATE UNIVERSITY

Math 285 Final Project
Terry Situ



Introduction

* A lot of autonomous vehicles are equipped with on-board digital cameras to capture pictures (data) of objects on the roads;
* I implemented and compared the methods learned in this semester. The following classifiers yields some better results:

1. Principal Component Analysis (PCA)+Local kMeans
2. Random Forests
3. Principal Component Analysis (PCA) + Quadratic Discernment Analysis (QDA)
4. PCA + Support Vector Machine (SVM) with Gaussian Kernel + One vs. All Extension

Data

CIFAR 10: 32 x 32 pixels x 3 Layers (Red, Green & Blue); Color Images

	# of Classes	Data Size
Training	10	50000
Test	10	10000

Algorithm

The Algorithm:
PCA + Local kMeans

- Applied Principal component Analysis (PCA) to both of the training and test data
  - Input: data set $X = [x_1 \ldots x_n]^T \in R^{n\times d}$ and integer s (with $0 < s < d$)
  - Output: compressed data $Y \in R^{n\times s}$
  - Steps:
    - Center data: $\widetilde{\boldsymbol{X}} = [x_1 \ldots x_n]^T - [m \ldots m]^T$ where $\mathbf{m} = \frac{1}{n}\sum x_i$;
    - Perform Singular Value Decomposition (SVD): $\widetilde{\boldsymbol{X}} = \boldsymbol{U\Sigma V^T}$, where $\boldsymbol{U}$ consists of the eigenvectors of $\boldsymbol{XX^T} \in \boldsymbol{R^{n\times n}}$, V consists of the eigenvectors of $\boldsymbol{X^TX} \in \boldsymbol{R^{d\times d}}$, and $\boldsymbol{\Sigma}$ consists of the square roots of the eigenvalues of either matrix (and zero).
    - Return: $\boldsymbol{Y} = \widetilde{\boldsymbol{X}}\,\boldsymbol{V}(:,\mathbf{1}:\boldsymbol{s}) = \boldsymbol{U}(:,\mathbf{1}:\boldsymbol{s})\boldsymbol{\Sigma}(\mathbf{1}:\boldsymbol{s},\mathbf{1}:\boldsymbol{s})$, where s is the top number of principal components.
- Use the k nearest points in each class to represent the corresponding class
- Perform $k$means classification based on the reduced data
  - Assign labels to test images based on the most similar centers

$$\hat{y} = argmin_y dist(\mathbf{x}, C_y)$$

in which $dist(x, C_y)$ represents the distance from a test point $\mathbf{x}$ to a training class C_y.

The Algorithm:
Random Forests

- Build a classification tree on a separate bootstrap sample of the training data
  - Bootstrap sample: random sample with replacement
- Every tree in the ensemble to randomly select features of the training data for its nodes

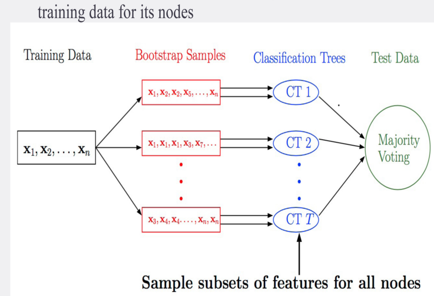


Photo Credit: Chen, Guangliang. "Lecture 8: Classification trees and ensemble learning". Math 285 – Spring 2016

The Algorithm:
PCA + QDA

- Applied Principal component Analysis (PCA) to both of the training and test data as usual
- Applied Quadratic Discriminant Analysis (QDA) classifier to the training data:
  - Model the distribution of each training class C_i by a pdf $f_i(\mathbf{x})$, and assign the label based on the posterior probability

$$i^* = argmax_i f_i(\mathbf{x})\pi_i;$$

  - For QDA, we use multivariate Gaussian distributions to estimate $f_i(\mathbf{x})$.

The Algorithm:
PCA + SVM Kernel + One vs. All

- Applied PCA to the training and test data as usual
- Construct the classifier for each class against the rest of training set
  - The "most clearly winning" class is adopted as the final prediction
- Applied SVM with Gaussian kernel
  - Solve the following quadratic program

$$\max_{\lambda_1,\ldots,\lambda_n} \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i\lambda_j y_i y_j k(\mathbf{x_i}, \mathbf{x_j})$$

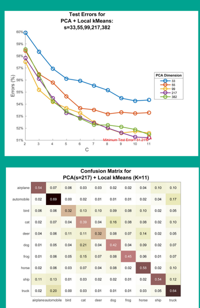
subject to $0 \le \lambda_i \le 0$ and $\sum \lambda_i y_i = 0$
where $k(\mathbf{x_i}, \mathbf{x_j})$ is the Gaussian kernel function

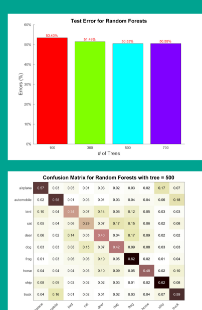
  - Classify new data $\mathbf{x}$ based on the following decision rule:

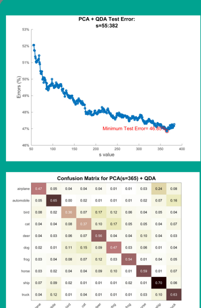
$$y = sgn(\sum \lambda_i y_i k(\mathbf{x_i}, \mathbf{x}) + b)$$

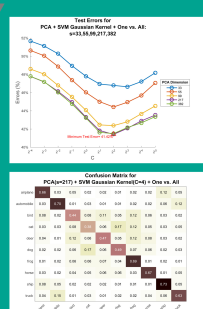
where b can be determined from any support vector with $0 \le \lambda_i \le C$.

Results









The Best Results from This Project

Test Error	Method
51.21%	PCA(s=217)+Local kMeans(k=11)
50.03%	Random Forests with tree = 500
46.85%	PCA(s=365)+QDA
41.42%	PCA(s=217)+SVM Gaussian Kernel(C=4)+One vs. All

Conclusions

- PCA + SVM method produces the lowest test error rate but the computation time is the longest
- Test Error is relatively high by using methods learned from this semester;
- The CIFAR 10 data only contains low resolution pictures;
- Should collect my own photos as test data in future studies.

Some outstanding research results

Test Error	Method
21.14%	Enhanced Image Classification With a Fast-Learning Shallow Convolutional Neural Network
20.40%	An Analysis of Single-Layer Networks in Unsupervised Feature Learning
17.82%	Convolutional kernel Network
3.46%	Fractional Max-Pooling

References

A. De la Escalera, J. M. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," Image and vision computing, vol. 21, no. 3, pp. 247–258, 2003.

Blundell, Heather, and Sarah M. Thornton. Object Classification for Autonomous Vehicle Navigation of Stanford Campus. Rep. Standford, 2015. Web.

Chen, Guangliang. "Lecture 8: Classification trees and ensemble learning". Math 285 – Spring 2016

Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images (pp. 32-33, Tech.).

Levinson, Jesse, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J. Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. "Towards Fully Autonomous Driving: Systems and Algorithms." 2011 IEEE Intelligent Vehicles Symposium (IV) (2011). Web.