SAPIENZA
UNIVERSITÀ DI ROMA

Dipartimento di Ingegneria Informatica, Automatica e Gestionale - Antonio Ruberti

Master in Artificial Intelligence and Robotics

Report for the project in Machine Learning

Multi-class image classification using logistic regression and one-vs-all algorithm

STUDENT:
Giovanni Murru

PROFESSOR:
Prof. Luca Iocchi

WINTER 2011

Abstract

Because of its several application fields, image classification is one of the most active topic of research in machine learning. The following report describes the use of the well known one-vs-all algorithm, which is normally used in the field of multi-class classification, for the classification of five different types of boats, and the absence of boats, in the contest of the ARGOS project [4].

Previous approaches used Artificial Neural Networks to solve the problem. Here in the following we demonstrate how it is possible to obtain a decent accuracy in classification using Opensurf feature extractor and the power of logistic regression with one-vs-all.

Contents

Introduction

The information is a constantly increasing flow of numerical data. The perception of an image can be described by a signal and every signal can be represented as numbers. The problem of understanding the patterns that may arise in these numbers is one of the main topics of discussion in the scientific community.

In the society of internet the quantity of images is exponentially increasing and the utility to classify or cluster them in a automated reasonable way is one of the most demanded topics in the innovative scientific fields of machine learning and pattern recognition.

This new active topic of research can help to solve disparate problems: from security field up to social networking tasks. For example an image classifier can be the autonomous tagger for facebook, which suggest the name of a friend when a new photo is uploaded on the social network. On the other hand in the security field, there's a desperate need to use efficiently data coming from surveillance cameras.

In this report we will show an attempt to find correct parameters for automatically identifying 6 different categories of images coming from surveillance cameras placed in the Grand Canal of Venice, part of the ARGOS (Automatic and Remote Grandcanal Observation System) project.[4]

The results are not excellent because of the lack of training data, but it is possible that having a bigger training set can help us to build a better classifier for these types of boats. The given training set was extended using translation, deformation techniques to obtain a homogeneous one with 100 examples per class.

We performed several experiments using directly the raw data of the images, and using a well-known image feature extractor named OpenSurf. Comparing the results underlined the necessity to use feature extractors, due to the highly non-linear nature of the images.

1 A Theoretical Review

Learning is about understanding how the future and the past are related. Supervised learning is probably one of the most common type of machine learning problems which consists in finding a function able to map some data in a meaningful way, using the help of a certain number of training examples.

Suppose you have a function $f(x)$, but you don't know what the function is in reality, what you know is just a set of x and a corresponding set of $f(x)$ values. We can define the set of these correspondences $[x, f(x)]$ as the *training set* which will help our supervised algorithm to understand the meaning of the function f.

One of the most common application of supervised learning is its discretized version, known as classification. In classification the possible outcome of $f(x)$ belongs to a finite set of values.

An example of classificator can be an algorithm able to group and name different categories of vehicles, given some information about them. In particular a possible example is a software that given the number of wheels, the price, and maybe some other general and relevant properties of the vehicles, it is able to classify them as cars, motorcycles or bicycles.

1.1 Classification using images

In the case of image classification, the only feautures we have are inside the image itself. Image however is an infinite source of data. However this infinite resource of data is not easy to manage. There are numerous techniques to extract meaningful features from images and use them to train a machine learning algorithm. In case of a high number of training examples it is also possible to use raw data coming from the images, as the luminance value of each pixel.

Figure 1.1: The problem of classification: identify three types of vehicles.

1.2 Logistic Regression

The core of the logistic regression algorithm is the logistic function, which, like probabilities, always takes on values between zero and one.

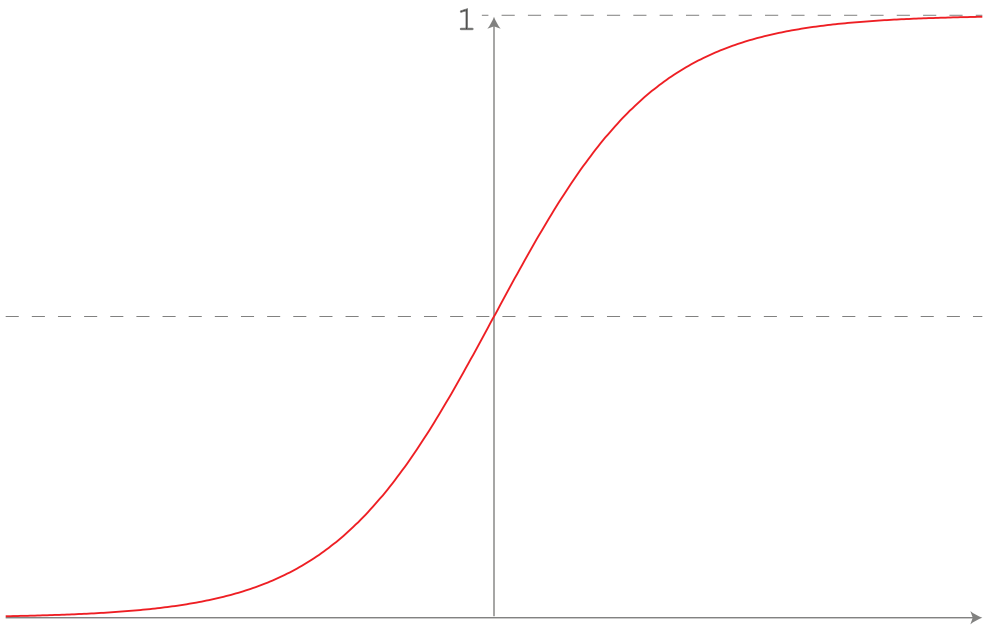


Figure 1.2: Logistic regression curve

$$g(z) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1.1}$$

The equation 1.1 is the logistic function, also known as sigmoid, whereθ represents the vector of parameters that the algorithm will try to optimize, and x is the vector of features.

The hypothesis output $g(z)$ can be interpreted as estimated probability that $y = 1$ on input x. It is evident that such type of function is useful for binary classification, for example a classifier of images informing if in the image there's a pedestrian or not. However we will see how the use of this function in the particular contest of one-vs-all algorithm will help us to solve multi-classification problems.

As in every classification problem we need to define a decision boundary, a contour that will serve as indicator for the decision making of the classification problem. In the case of logistic regression we can define this border as the mid point 0.5, indicated by a dashed line in figure 1.2.

In fact, suppose we predict $y = 1$ for $g(z) \geq 0.5$ and $y = 0$ if $g(z) < 0.5$, this is equivalent to say that:

$$\theta^T x \geq 0 \quad \text{if} \quad y = 1 \tag{1.2}$$

$$\theta^T x < 0 \quad \text{if} \quad y = 0 \tag{1.3}$$

The value of y is the value of the class. The $g(z)$ is nothing else that the hypothesis made to classify the items. The unknown in this hypothesis are the parameters θ and the main problem of optimization is how to choose these parameters in order to make a correct classification.

As a matter of fact the hypothesis is just a function of $z = \theta^T x$.

$$h_\theta(x) = g(\theta^T x) \tag{1.4}$$

The decision boundary can assume different shapes, even complex ones, based on the equation describing it. In order to find an optimization of the parameters a cost function is defined.

Suppose we have m training examples $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})\}$, where $x^{(i)}$ is a $(n+1)$ dimensional vector containing n features (the first element is 1), and $y^{(i)}$ is a binary value $\in \{0, 1\}$.

We define the logistic regression cost function as:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)}) \tag{1.5}$$

where the value of Cost depends on the value of y. In particular:

$$\text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \begin{cases} -log(h_\theta(x)) & if \quad y = 1 \\ -log(1 - h_\theta(x)) & if \quad y = 0 \end{cases} \tag{1.6}$$

To operation performed in order to find the optimal parameters θ is called optimization and consists in finding those parameters that minimize the cost function $J(\theta)$.

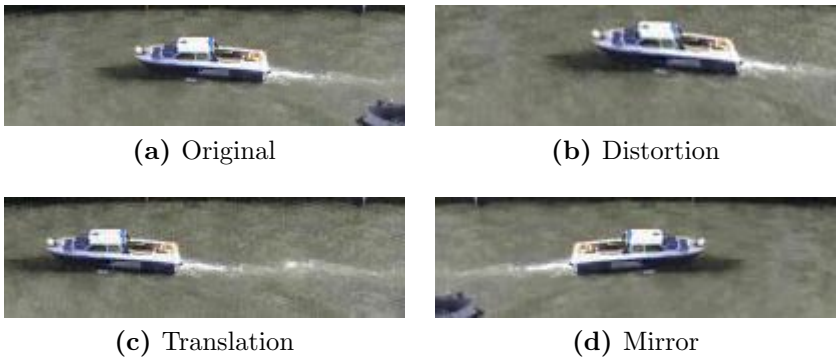
A classic method used to find the parameters that minimize the cost function is gradient descent, however there exist advanced optimization techniques developed by mathematicians which perform much better and for this reason were preferred in the optimization objective. In particular we used a function to minimize a continuous differentiable multivariate function developed by Carl Edward Rasmussen, a researcher and professor at Cambridge University Engineering Dept. [7].

One of the advantages of using these advanced optimization functions is the fact that we don't need to manually pick a learning rate α. Furthermore usually the optimization is faster than the normal gradient descent.

As anticipated before, multiclass classification can be reduced to the simple case of binary classification. In particular one-vs-all algorithm trains a logistic regression classifier h_θ^i for each class i that we have to predict, then on new input x the prediction step is to choose the class i that has the highest hyopothesis h_θ^i.

2 Implementation

One of the main problems of the implementation was the lack of training examples. To compensate the scarcity in the training set, an artificial extension using operations of translation, distortion and mirroring in some of the images, has been applied, resulting in a data set with 100 examples per class.



(a) Original **(b)** Distortion

(c) Translation **(d)** Mirror

Figure 2.1: Example of images obtained by artificial extension

The classifier had to deal with a set of images representing boats at different angles and different light conditions. In particular the dataset includes 6 different categories of images:

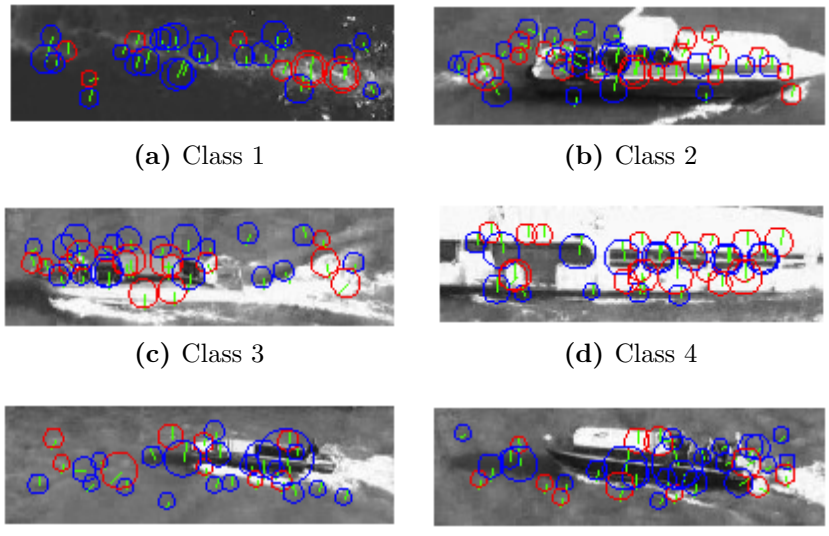
1. No boats [1]
2. Big motorboats
3. Taxi boats
4. Water bus (Vaporetti)
5. Small motorboats
6. Other boats

As first approach we tried to run the algorithm using the raw data of a grayscale and scaled version of the original dataset, obtaining an accuracy around 50% in the test and validation set. However the automatic validation of lambda did not find the optimal one and the parameter was manually chosen. Furthermore this approach is limited by the fact that images includes also detail and portions of something which is not a boat, and by the high dimensionality of features.

[1] in this class are included photographs of the grandcanal without boats, most of them showing portion of water only.

Hence we decided to use a feature extractor, namely the OpenSurf feature extractor [6], which helped us to reduce the dimensionality of the features localizing only the important ones.

Each feature has a descriptor of 64 numbers describing its properties. After several tests we decided to take 13 features per image. One of the problem was how to choose these 13 features. We will see in the next section how the problem was solved.



(a) Class 1 **(b)** Class 2

(c) Class 3 **(d)** Class 4

(e) Class 5 **(f)** Class 6

Figure 2.2: Example of feature extraction obtained using Opensurf on the six classes

3 Experimental results

Several experiments with different parameters were accomplished in order to validate the results, obtaining an accuracy between 60% and 70%, with peeks of 75% in lucky initializations.

One of the problem of extracting features was to choose a metodology to select part of them based on a valid criteria. The first approach was to use a reference image for each of the class and extract the features from the other images based on the class they belonged. This approach, as expected, resulted in a perfect working classifier with 100% of accuracy in each of the test, training and validation sets. This experiment was performed just to understand if the algorithm was working correctly.

In a second attempt we decided to extract features using multiple reference images. In practice for each image we extracted features a number of times equal to the number of classes that we had to classify; then we chose the set of features nearest to the image.

In a third experiment we created a number of artificial images merging the class

Set	Accuracy	Class	TP	FP	FN	TN	P	R	F1
T		No boats	70	0	0	350	1.0	1.0	1.0
R		Big motorboats	70	0	0	350	1.0	1.0	1.0
A	99.76%	Taxi boats	70	0	0	350	1.0	1.0	1.0
I		Water bus	69	0	1	350	1.0	0.986	0.993
N		Small motorboats	70	1	0	349	0.986	1.0	0.993
		Other boats	70	0	0	350	1.0	1.0	1.0
V		No boats	15	3	0	72	0.833	1.000	0.909
A		Big motorboats	9	3	6	72	0.750	0.600	0.667
L	74.44%	Taxi boats	11	6	4	69	0.647	0.733	0.688
I		Water bus	12	3	3	72	0.800	0.800	0.800
D		Small motorboats	11	2	4	73	0.846	0.733	0.786
		Other boats	9	6	6	69	0.600	0.600	0.600
T		No boats	15	1	0	74	0.938	1.000	0.968
E		Big motorboats	10	3	5	72	0.769	0.667	0.714
S	73.33%	Taxi boats	8	4	7	71	0.667	0.533	0.593
T		Water bus	10	3	5	72	0.769	0.667	0.714
		Small motorboats	14	7	1	68	0.667	0.933	0.778
		Other boats	9	6	6	69	0.600	0.600	0.600

Table 3.1: Data evaluation of a run using a single image as reference for extracting the feautures.

Figure 3.1: Reference image used to extract the features

reference images in different ways: the best performing one is shown in figure 3.1. Then we used this artificial image as unique reference for extracting the features for each of the boats' classes.

In the table we can see the results of a algorithm run. One-vs-all performed well enough and we believe that with a better definition of the classes and a bigger number of training examples a better classifier might be created.

From the result we can notice how some classes performed much better than others. As a matter of fact the best classified class is the *No boats* class. This is probably due to the fact that this class is composed by images very similar with each other, usually representing empty portion of the grand canal with water only. The achieved F1 score for this class is of 0.91 in the validation set with 15 true positives, 72 true negatives and only 3 false positives and 0 false negatives. F1 score in the test set is even better with a value 0.97. Not surprisingly the worst

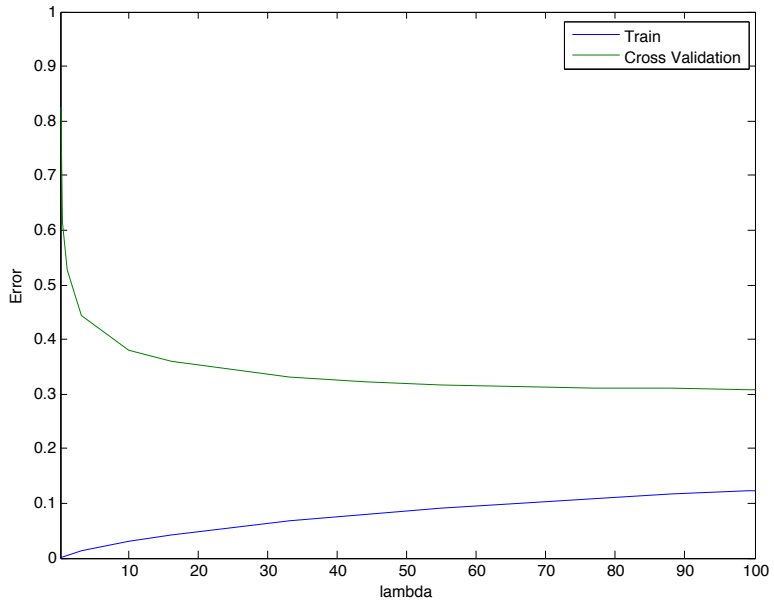


Figure 3.2: Plot of validation and train error in function of the regularization parameter λ

class in performance is *Other boats* which include photographs of different type of boats that can be barely classified in the same cluster even by a human operator not expert in the sector. F1 score for this class was 0.6 in both validation and test sets.

The algorithm suffered of high variance as highlited by the plot in figure 3.2; however increasing the regularization parameter λ and trying a smaller set of features didn't help to improve the results. With a good probability it is possible that a higher number of training examples would help to reach the goal.

At the beginning of the algorithm the data set is randomly partitioned in 3 parts, maintaining an equal number of samples for each class:

- Training set: 70%
- Validation set: 15%
- Test set: 15%

After the partition the sets are normalized and shuffled again. Then the best regularization parameter is searched and located basing the decision on the validation error. Once the best λ is found, it is used to compute the parameters θ using the One-vs-all algorithm iteratively and stopping when the cross validation error stops to decrease. We can see in figure 3.3 the plot of the learning curve obtained during

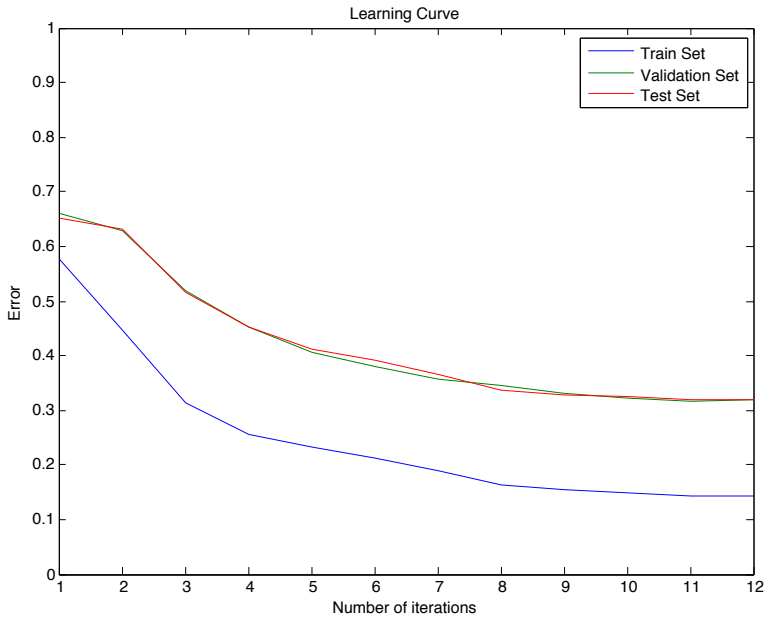


Figure 3.3: Learning curve for validation, train and test sets

this operation. Note that the errors in the plot are computed without using the regularization parameter.

In conclusion, considered the lack of training examples we think the one-vs-all with logistic regression performed as expected. A possible idea to improve the results can be to crop the photos including precisely only the boat and train the logistic classifier with that information. Furthermore it is necessary to collect more training examples, then it is hopefully possible to iteratively use sliding windows of different size to perform an online scan from each frame coming from the camera, detect the different kinds of boats and hence classify them.

List of Figures

List of Tables

References

[1] Luca Iocchi, *Slides Machine Learning*, 2011

[2] *Machine Learning*, Tom Mitchell, McGraw Hill, 1997

[3] Andrew Ng, *Stanford ML-Class*, 2011

[4] *http://www.argos.venezia.it*

[5] Weifeng Liu, Jose C. Principe, and Simon Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction.* John Wiley & Sons, Inc., Publication

[6] *http://www.chrisevansdev.com/computer-vision-opensurf.html*

[7] *http://learning.eng.cam.ac.uk/carl/*