Statistical Methods for Machine Learning
Final Exam Assignment
Recognition of Traffic Signs

Christian Igel and Kim Steenstrup Pedersen

March 2011

This is the final exam assignment on the course *Statistical Methods for Machine Learning*, block 3 2011, at the University of Copenhagen. It is based on the full course curriculum as stated in the lectures schedule in the Absalon system. The assignment is centered around a real world pattern recognition task, namely the recognition of traffic signs by a driver assistance system.

This assignment must be made and submitted **individually**. However, it is acceptable (and we encourage) that you discuss the solution of the assignment with your fellow students. It is also acceptable (and we encourage) to use bits and pieces of your solutions and the hand-out code from the previous assignments.

Your solution to this assignment will be graded using the 7-point scale and will be the final grade for the course. To obtain the best grade of 12, you must fulfill all the course learning objectives (see below) at an excellent level. In terms of the questions in this assignment, this means that you have to answer all questions with non or only a few mistakes or parts missing. To obtain the passing grade of 02 you need to fulfill the learning objectives at a minimum level, which means you have to make a serious attempt at solving the central questions in the assignment (but not necessarily all) with some mistakes allowed.

The deadline for this assignment is **Thursday, April 7 2011**. You must submit your solution electronically via the Absalon home page. Go to the assignments list and choose this assignment and upload your solution prior to the deadline.

Solution format

The deliverables for each question are listed at the end of each question. The deliverable "description of software used" means that you should hand in the source code you have written to solve the problem. If you have used a tool to solve the problem, this tool should be described and reasons for the particular choice should be given.

Thus, a solution should contain:

- A PDF document showing your results and giving detailed answers to the questions. If relevant, this may include graphs and tables with comments (**max. 10 page of text including figures and tables**). Use meaningful labels, captions, and legends. Do **not** include your source code in this PDF file. You will be graded mainly on the basis of this report.
- Your solution code (Matlab / R / Python scripts or C / C++ code) with comments about the major steps involved in each question. The code must be submitted in its original format (e.g., in `.m` or `.R` file format – not as PDF files). Use meaningful names for files, constants, variables, functions and procedures etc. Add comments to the code to make it more readable.
- Your code should also include a README text file describing how to compile (if relevant) and run your program, as well as a list of all relevant libraries needed for compiling or using your code. If we cannot make your code run we will consider your submission incomplete.

Learning objectives

The grade will be based on a judgement of how well you fulfill the following learning objectives:

1. Recognize and describe possible applications of machine learning for pattern recognition and data mining.
2. Explain, contrast and apply basic Bayesian probability theory for modeling stochastic data, including both parametric and non-parametric representations.
3. Explain and contrast the concept of supervised and unsupervised learning.
4. Explain the concepts of classification and clustering.
5. Identify, explain and handle the common pitfalls of machine learning.
6. Describe and apply linear techniques for classification
7. Implementation of selected machine learning techniques.
8. Use software libraries for solving machine learning problems.
9. Visualize and evaluate results obtained with a machine learning method.
10. Compare, appraise and select methods of machine learning for solving specific problems of pattern recognition and data mining.

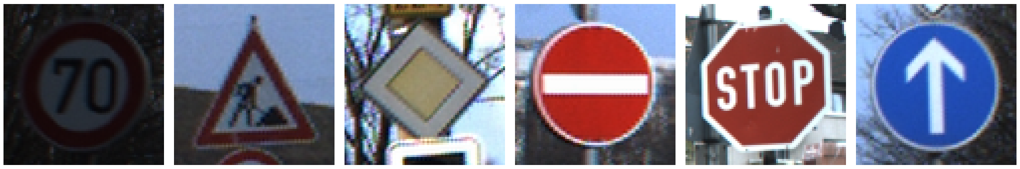

Figure 1: Examples from the traffic sign data set.

The Traffic Sign Recognition Data

We consider the "German Traffic Sign Recognition" benchmark data available from `http://benchmark.ini.rub.de`, which is currently used in a machine learning competition at the IEEE International Joint Conference on Neural Networks (IJCNN), the largest neural network conference.

Recognition of traffic signs is a challenging real-world problem of high industrial relevance. Traffic sign recognition can be viewed as a multi-class classification problem with unbalanced class frequencies, in which one has to cope with large variations in visual appearances due to illumination changes, partial occlusions, rotations, weather conditions, etc. However, humans are capable of recognizing the large variety of existing road signs with close to 100 % correctness – not only in real-world driving situations, which provides both context and multiple views of a single traffic sign, but also when looking at single images. Now the question is how good a computer can become at solving this problem. The data set consists of traffic sign images of different sizes, see figure 1 for examples.

We consider already preprocessed images. The data relevant for the exam is contained in `GTSRB_Training_Features_HOG` and `GTSRB_Online-Test-HOG-Sorted`. These data have been produced by transforming the images into a reduced size, fixed length, real-valued feature vector, which is nice for machine learning. The features are histograms of oriented gradient (HOG). There are 43 classes, each class corresponds to one type of traffic sign. Reliable ground-truth data were obtained by semi-automatic annotation. We refer to a point in the data set also as an *image*. To solve the following exam, it is helpful to understand how the images were generated: Videos (*tracks*) were shot from a driving car and the traffic signs were extracted from these videos. This implies that the data set contains several images of the same *physical traffic sign* taken from subsequent frames of the video.

For the exam, we use the data in the folders named `HOG_01` (feel free to play around with other data sets from the benchmark website and even try your own features generated from the raw image data). We refer to this data set as $\mathcal{D}$.

The file folders containing the training and test data are structured as follows. First, there is one directory per class. Each directory contains one file with annotations (`GT-ClassID.csv`) and the training images. Training images are grouped by tracks. Each track contains 30 images (also referred to as instances) of the same single physical traffic sign (from different angles and distances).

Auxiliary code (e.g., for reading in the data) and details about the data can be downloaded from `http://benchmark.ini.rub.de`.

Note that the data requires a considerable amount of disk space and that some of the algorithms require a considerable amount of time when applied to the full data set.

The Exercises

Question 1 (data understanding and preprocessing)**.** Download and extract the training data:

`http://benchmark.ini.rub.de/Dataset/GTSRB_Training_Features_HOG.zip`

Plot a histogram showing the distribution of class frequencies (i.e., for each of the 43 traffic signs plot the number of observations in the training data set divided by the total number of observations in the training data set).

Deliverables: description of software used; single histogram plot

Question 2 (principal component analysis)**.** Perform a principal component analysis of $\mathcal{D}$. Plot the eigenspectrum. How many components are necessary to "explain 90 % of the variance"? Visualize the data by a scatter plot of the first two principal components. Use different colors for the different shapes in the plot.

Table 1 provides the mapping from class index to shape.

Deliverables: description of software used; plot of the eigenspectrum; number of components necessary to explain 90 % of variance; scatter plot for first two principal components with different colors indicating the five different shapes

Question 3 (clustering)**.** Perform 4-means clustering of $\mathcal{D}$ (feel free to play around with the number of clusters). Plot the cluster centers projected to the first two principal components. That is, add the cluster centers to the plot from the previous exercise. Briefly discuss the results: Did you get meaningful clusters?

Deliverables: description of software used; one plot with cluster centers; short discussion of results

Question 4 (overfitting)**.** John Langford, who is "Doctor of Learning at Yahoo Research", maintains a very interesting blog (web log). Read the very true blog entry: "Clever methods of overfitting," `http://hunch.net/?p=22`, 2005. Discuss if and how the different types of overfitting can occur when applying machine learning techniques to $\mathcal{D}$. Ignore the last type of overfitting. You need not discuss issues related to reviewing of scientific papers (still, it is good to keep them in mind).

Deliverables: short discussion addressing the first 10 "methods of overfitting" listed in the blog entry

Table 1: Mapping from class number to shape of the encoded traffic sign.

class index	shape
00000	Round
00001	Round
00002	Round
00003	Round
00004	Round
00005	Round
00006	Round
00007	Round
00008	Round
00009	Round
00010	Round
00011	Upwards pointing triangle
00012	Diamond
00013	Downwards pointing triangle
00014	Octagon
00015	Round
00016	Round
00017	Round
00018	Upwards pointing triangle
00019	Upwards pointing triangle
00020	Upwards pointing triangle
00021	Upwards pointing triangle
00022	Upwards pointing triangle
00023	Upwards pointing triangle
00024	Upwards pointing triangle
00025	Upwards pointing triangle
00026	Upwards pointing triangle
00027	Upwards pointing triangle
00028	Upwards pointing triangle
00029	Upwards pointing triangle
00030	Upwards pointing triangle
00031	Upwards pointing triangle
00032	Round
00033	Round
00034	Round
00035	Round
00036	Round
00037	Round
00038	Round
00039	Round
00040	Round
00041	Round
00042	Round

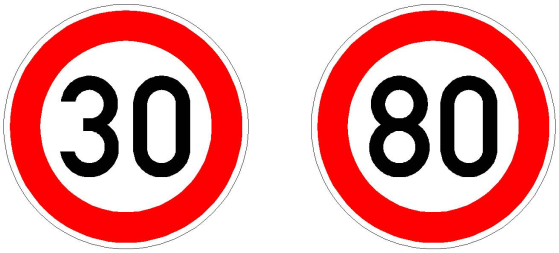

Figure 2: Sign classes 0001 and 0005.

Question 5 (binary classification)**.** Generate a binary classification problem from $\mathcal{D}$ by considering only the signs 00001 and 00005 (see figure 2).

Apply a linear and a non-linear classification method (picking from the methods presented in the course) to distinguish between those two classes. Use cross-validation to evaluate your models and report your results.

For cross-validation, partition the training data set *randomly* into five subsets. However, in our particular example the partitioning is a bit tricky. You should ensure that all 30 instances of the same physical traffic sign are in the same subset. Otherwise, you get overoptimistic results (why?).

Deliverables: description of software used; arguments for your choice of classification methods; a short description of how you proceeded and what results you achieved, in particular how you partitioned the data and how you computed the cross-validation test errors of the linear and non-linear classifiers

Question 6 (multi-class classification)**.** Use a linear and a non-linear classification method (picking from the methods presented in the course) for classifying the 43 traffic sign classes. Apply cross-validation to evaluate your models and report your results.

Deliverables: description of software used; arguments for your choice of classification methods; a short description of how you proceeded and what results you achieved

Question 7 (generalization)**.** Download the "German Traffic Sign Recognition" benchmark *test* data:

`http://benchmark.ini.rub.de/Dataset/GTSRB_Online-Test-HOG-Sorted.zip`

We consider the data in the subfolder `HOG_01`.

Evaluate the classifiers you built in the previous exercises 5 and 6 on these test data.

Deliverables: report of the results; comparison of accuracies on training and test data

Question 8 (linear regression)**.** Apply linear regression to the binary as well as to the multi-class classification problem. Use the class number as the target for regression. Compare your results of this "abuse" of regression for solving a classification task to those achieved in exercises 5 and 6.

Deliverables: description of software used; discussion of results

Question 9 (priors)**.** So far, we have consider the recognition of traffic signs, which requires successful detection of traffic signs in images. This is usually done in a multi-stage process. First, regions of interest (ROIs) are extracted from the input images. Let us assume that we know that 1 in 100000 ROIs is a traffic sign. Second, these ROIs are classified whether they contain a sign (positive class, encode by +1) or not (negative class, encode by -1). Assume that we have a pretty good classifier: If the ROI contains indeed a traffic sign, then it correctly answers +1 with a probability of 99 %. If the ROI belongs to the negative class, then it correctly answers -1 with a probability of 99 %. Given 10^6 ROIs, how many false positives can we expect? That is, how many ROIs containing no traffic sign are wrongly classified as being a sign?

Deliverables: calculation leading to the expected number of false positives