CSE 592: Advanced Topics in Computer Science
Principles of Machine Learning (Fall 2011)
Department of Computer Science
Stony Brook University

Homework 3: On Nearest Neighbors and Decision Tree Classifiers

Out: *Thursday, November 17* **Due**: *Monday, November 21*

Recognizing Handwritten Digits Automatically

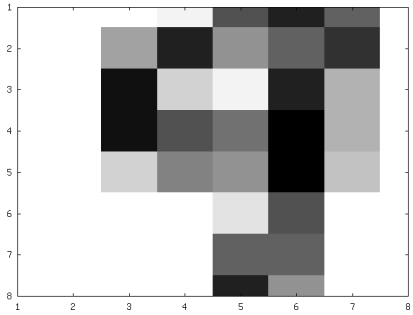
In this part of the homework you will apply the nearest-neighbor machine-learning technique to a stylized version of the problem of optical recognition of handwritten digits.

The data set of examples was created by pre-processing or transforming 32x32 bitmap images of hanwritten digits (i.e., each image was originally represented as a 32x32 matrix of pixels, each pixel taking value 1 or 0 corresponding to a black or white pixel, respectively). The result of the transformation is an 8x8 "grayscale" image where each pixel takes an integer value from 0 to 16. [1]

The following is an example for the digit 9:

Original Image

Transformed Image



For each transformed image, each pixel corresponds to an attribute taking one of 16 values. Thus, the data has 64 attributes. Although each attribute is an integer from 0 to 16, treat each attribute as real-valued. There are 10 classes corresponding to each of digit.

You are provided four data files:

- `optdigits_tra.dat` contains the original training data;
- `optdigits_tra_trans.dat` contains the pre-processed training data;
- `optdigits_trial.dat` contains an example of each digit from the original validation data set ;
- `optdigits_trial_trans.dat` contains preprocessed examples from `optdigits_trial.dat`

Each file is composed of one data example per line. Each line of the `..._trans.dat` files contain 65 integers separated by a single empty space. The first 64 integers correspond to the values of each of the attributes (i.e., a number from 0 to 16) and the last integer is the example class label (i.e., the corresponding digit $0, 1, \ldots, 9$). The case is similar for the other files but now each line contains 1025 integers separated by a single space. The first 1024 correspond to the values of a bitmap of

[1] Pre-processing is done to "normalize" the data to help correct for small distortions and reduce dimensionality. The resulting images provide a good approximation of the original images for classification purposes. Please visit `http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits` for more information about this data set.

the image, where each consecutive sequence of 32 values, starting with the value in the first column, corresponds to a row of black (0) and white (1) pixel values of the image of the handwritten digit. The value of the digit is given as the last element of the 1025 vector/row. A proper reshaping of the vector composed of the first 1024 binary values in the file-row will produce a 32x32 black-and-white image. The training and trial data sets have 1934 and 10 examples, respectively.

Applying k-Nearest Neighbors Classification

You are asked to implement the technique of k-nearest neighbors classification using Euclidean distance as the distance metric and apply it to the data set of optical handwritten digits described above. *As a tie-breaking rule during classification, select the lowest digit as the class label (i.e., if there is a tie between* 3 *and* 7*, pick* 3 *as the label).*

Perform the following steps:

1. For each transformed image example in the trial data set, identify the 3-nearest neighbors *in the transformed space* using the training data file of transformed images .

2. Display the corresponding *original 32x32 pixels, B&W images* of the 3-nearest neighbors of each of the 10 exemplars in the trial data set. Display the images as a row, starting with the exemplar itself, and followed by the 3-nearest neighbors, *in increasing order of Euclidean distance.*

Space Shuttle Autolanding Controller using Decision Trees

In this homework you are asked to learn a decision tree to represent "comprehendible rules for determining the conditions under which an autolanding would be preferable to manual control of the spacecraft." [2] The data set was obtained from the UCI Machine Learning Repository [Frank and Asuncion, 2010]. (Please visit `http://archive.ics.uci.edu/ml/datasets/Shuttle+Landing+Control` for more information on this particular data set.)

The binary target concept is the advise on using manual or automatic control (`AUTO`). The controller decision rule is based on six (6) attributes, labeled as `STABILITY`, `ERROR`, `SIGN`, `WIND`, `MAGNITUDE`, and `VISIBILITY`. The target concept and the attributes can be thought as either binary or ordinal types, with values (and integer mappings) given by the following table.

attributes	**values**
`STABILITY` (1)	`stab` (1), `xstab` (2)
`ERROR` (2)	`XL` (1), `LX` (2), `MM` (3), `SS` (4)
`SIGN` (3)	`pp` (1), `nn` (2)
`WIND` (4)	`head` (1), `tail` (2)
`MAGNITUDE` (5)	`Low` (1), `Medium` (2), `Strong` (3), `OutOfRange` (4)
`VISIBILITY` (6)	`yes` (1), `no` (2)
concept	**values**
`AUTO`	`noauto` (1), `auto` (2)

[2] `http://archive.ics.uci.edu/ml/machine-learning-databases/shuttle-landing-control/shuttle-landing-control.names`

The original data set of examples consists of only 16 examples, including "don't care" conditions. The "don't care" conditions in some examples of the original data set have been expanded into all possible specific assignments yielding a data set of 253 complete specific examples. [3] Each line in the file `shuttle_ext_unique.dat` corresponds to each of those 253 examples. For instance, the following data example

```
AUTO=auto, STABILITY=xstab, ERROR=MM, SIGN=pp, WIND=head, MAGNITUDE=OutOfRange,
                                VISIBILITY=no
```

is represented in the data set file as the following space-separated list of seven (7) integers,

```
2  2  3  1  1  4  2
```

where the first integer corresponds to the target concept value (i.e., the output) and the last six integers correspond to the attribute values in order (i.e., the input).

Learning a Decision Tree for Space Shuttle Autolanding Controller

You are asked to implement the decision-tree machine-learning technique (see handout of Chapter 18 of Russell and Norvig's AI book for more information), using `noauto` as default, and apply it to the space shuttle autolanding controller data.

Perform the following steps:

1. Apply the decision-tree learning algoritm to the training data, using *information gain* as the measure to select attributes.
   (a) Draw the resulting tree.
2. Repeat the previous step, this time using *gain ratio* as the attribute-selection measure.
   (a) Compare and contrast the two resulting trees.

What to Turn In

You need to submit the following.

1. A **written report** (*in PDF*) that includes the following.
   (a) An 10x4 array of images where each row corresponds to each of the 10 handwritten digits example images, provided as a vector in the file `optdigits_trial.dat` (in the first column), followed by the corresponding images of the 3-nearest neighbors (in columns 2-4). The images for the nearest neighbors are provided as vectors in the file `optdigits_train.dat`. (Please see the first part of the homework for a description of how to transform the vector representation of an handwritten digit into an black-and-white image.)

[3] Note that there are only 3 cases missing for the input data to cover the whole feature space.

(b) The drawings of the two decision trees you found for the shuttle autolanding controller data along with a brief statement comparing and contrasting the two trees. The internal nodes, branches and the leafs of both trees should be properly and clearly labeled with the corresponding attribute, attribute values and output/class, respectively.

Please submit the report both electronically and as a hard-copy.

2. All your **code and executable** (as a tared-and-gziped compressed file), with instructions on how to run your program. A platform-independent executable is preferred; otherwise, also provide instructions on how to compile your program. Please use standard tools/compilers/etc. generally available in most popular platforms. *Please submit the report both electronically.*

Please do the electronic submissions by uploading the material to corresponding assignment in Blackboard.

Collaboration Policy: *It is OK to discuss the homework with your peers, but each student must write and turn in his/her own report, code, etc. based on his/her own work.*

References

A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.