# Sugar beets project

## Task:

Image classification. Tell if the plant in the image is a beet or a weed, based on geometrical and/or colour data. You are not supplied with the raw image, only features extracted from the image (the features are described below).

## Data

You are given a file, *trainBeets.mat*, which contains the variables *Xtrain* ($450 \times 19$), *Ytrain* ($450 \times 1$), *Xtest* ($137 \times 19$), and *varnames* ($1 \times 19$).

The first matrix, *Xtrain*, contains the input data for the training set. The second matrix (vector), *Ytrain*, contains the output data for the training data (1 if the plant in question is a sugar beet, 0 if it is a weed). The third matrix, *Xtest*, contains the inputs for the test data. You will use this to produce a test output file that you must hand in at the end of the project. The fourth matrix (vector), *varnames*, contains the names of the variables in the 19 columns (6 color variables, 7 "shape" variables, and 6 statistical moments which are invariant to scale, translation and rotation). These are listed in Table 1.

The "statistical moments" are functions which describe the shape of the intensity distribution over the image, e.g. is it symmetric or skewed, wide or narrow, etcetera. That is, a 2D greyscale image is likened with a probability distribution in two dimensions, where the intensity in each pixel is likened with the probability for that pixel coordinate. Statistical moments are very similar to the mean value, the standard deviation, and so on.

## Steps and subgoals

1. Get acquainted with the data. Plot it and try to get a feel for the possible relationships between input and output. Try some different variable transformations (e.g. Box-Cox transformations) and see if this improves the discrimination power of the variables. The discrimination power can be measured using a Fisher-index measure.

2. Construct a quadratic Gaussian classifier for the problem, using all the variables, transformed if you want, and estimate the generalization error.

3. Prune the Gaussian classifier by successively removing the variable that results in the least degradation of the generalization error, until the degradation is significant. Note the classification error (generalization).

4. Construct a $k$-nearest neighbor ($k$NN) classifier, trying $k \in \{1, 3, 5\}$. Optimize the input variable set with respect to the generalization error (i.e. choose the set of input variables that maximize the generalization error).

| # | Name | Description |
|---|---|---|
| 1 | *greanmean* | The mean value, over the whole plant, of the normalized green color $g = G/(R + G + B)$. |
| 2 | *bluestd* | The standard deviation, over the whole plant, for the normalized blue color. |
| 3 | *area* | The area of the plant. |
| 4 | *perimeter* | The perimeter of the plant. |
| 5 | *compactness* | area/perimeter. |
| 6 | *bluemean* | The mean value, over the whole plant, of the normalized blue color. |
| 7 | *redstd* | The standard deviation, over the whole plant, for the normalized red color. |
| 8 | *elongation* | area/("thickness")$^2$. |
| 9 | *redmean* | The mean value, over the whole plant, of the normalized red color. |
| 10 | *greenstd* | The standard deviation, over the whole plant, for the normalized green color. |
| 11 | *solidity* | area/("convex area"). |
| 12 | *moment1* | (see below) |
| 13 | *formfactor1* | A measure of how much "plant mass" there is in the center in relation to how much "plant mass" there is in the perifery. |
| 14 | *convexity* | ("convex perimeter")/perimeter. |
| 15 | *moment4* | (see below) |
| 16 | *moment6* | (see below) |
| 17 | *moment5* | (see below) |
| 18 | *moment3* | (see below) |
| 19 | *moment2* | (see below) |

Table 1: The variables in the sugar beet data set.

5. Construct a multilayer perceptron (MLP) model using the best inputs for the Gaussian classifier. Optimize the number of hidden units (one hidden layer) with respect to the generalization error.

6. See if you can improve the MLP model by adding or removing any variables. Optimize the number of hidden units for the final model. Note the classification error (generalization).

7. Produce a linear SVM model, using all the variables, and optimize it with respect to $C$ (using cross-validation). Successively remove variables and see if this improves (or does not deteriorate) the performance. Optimize the $C$ value for each set of variables.

8. Produce a nonlinear SVM mocel (Gaussian kernel), using the best set of variables found for the MLP model (item 6 above). Optimize it with respect to the values of $C$ and $\gamma$.

9. Produce a test output file from your best Gaussian classifier. best $k$NN classifier, best MLP classifier, your best linear SVM classifier, and your best nonlinear SVM classifier. Estimate the expected test error and hand in the test files to me.

   **Note: The test files must be submitted to me no later than 48 hours before your oral presentation.**

10. Write a report. You are given five references that describe projects similar to the sugar beet plant detection problem: *ageng2000.pdf*, *exjobma.pdf*, *ICAME96.pdf*, *tomato.pdf*, and *YangEtAl2000.pdf*. Relate the results you get with the results reported by others on similar problems. If you want, then you can also consult the Halmstad University Master Thesis (E) that studied this data set.

## Report and presentation of results

You will present the results from your project in two ways: (1) A written report where the main conlusions are presented together with figures and tables supporting your conclusions. (2) An oral presentation, of about 20 minutes, to your course colleagues.

The report should be about 10 pages, including figures and tables, and should contain the elementary report constituents:

- Introduction (brief presentation of problem, 1 page)
- Methodology (brief listing of methods, 1 page)
- Data (presentation of your data set with important observations, 1-2 pages)
- Results (4-5 pages)

- Discussion (1 page)

The report writing should not take much more than one full day, since you are two persons sharing the work.

When you are finished with your report, and it has been accepted, then you should produce a file with it, and pack it together with your dataset and other important parts of your project (like MATLAB M-files). The idea being that someone else could unpack it and repeat the main steps in your analysis without rewriting everything.