Week 3:
Linear Classification Models

Anikó Ekárt

Module CS4730: Machine learning

Learning outcomes

In this unit we studied linear classification models. In this practical, we shall deepen our understanding by applying the perceptron and support vector machines and interpreting and comparing the results.

Instructions

Download the dataset `haberman.csv`.

We shall build linear classification models for this dataset, then evaluate and compare them.

You are encouraged to use Python Jupyter Notebooks for your work. This is for two main reasons: (1) the fact that you can print out intermediate results and execute code in smaller blocks will help you work through the tasks more efficently and effectively and (2) it is also the format that we shall be asking you to submit your portfolio tasks in.

Task 1

The dataset, `haberman.csv`, [1] contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. There are 306 instances in the dataset, shown as rows in the file. The columns represent:

1. Age of patient at time of operation (numerical)
2. Patient's year of operation ($year - 1900$, numerical)
3. Number of positive axillary nodes detected (numerical)
4. Survival status (class attribute): 1 = the patient survived 5 years or longer and 2 = the patient died within 5 years.

Divide the dataset into training and testing sets, by using the data for years 1958-1965 (229 instances) for training and 1966-1969 (77 instances) for testing.

[1] Donated to UCI repository by Tjen-Sien Lim, used by Haberman, S. J. (1976). Generalized Residuals for Log-Linear Models, Proceedings of the 9th International Biometrics Conference, Boston, pp. 104-122.

Apply a perceptron using `sklearn.linear_model.Perceptron`.

Use `sklearn.metrics.confusion_matrix` to obtain the confusion matrix of your classifier.

Use `sklearn.metrics.classification_report` to print out a report on precision, recall, f1-measure for both the training data and the testing data.

Experiment with different parameter settings in a systematic way, to see if you can improve upon your first model (you can consider here all of confusion matrix, precision, recall, f1-measure on both training and testing data).

Discussion. *For each solution, compare the results on training and testing data. What are the differences? Can you explain them? How did you decide which model is better? What parameters led to the best model?*

Task 2

For the same dataset and same division into training and testing data, apply linear Support Vector Classification using `sklearn.svm.LinearSVC` or `sklearn.svm.SVC` with the `kernel` parameter set to `'linear'`.[2]

Discussion. *Consider the same aspects as for Task 1. Did SVC lead to a better or worse model? Is your top solution sufficiently good in practice? Is your methodology for finding the best model generalizable to other problems?*

[2] We are focussing on linear classification this week, we shall study non-linear models in week 4.