Sample proposal L2 Mini Projects

Analysis of the Iris Data

Team name/project: PROF (Iris)
Group name: M
Team members: Isabelle Guyon < iguyon@lri.fr>, Zhengying Liu < zhengying.liu@lri.fr >, Balthazar Donon balthazar.donon@gmail.com, Adrien Pavao adrien.pavao@u-psud.fr
Challenge URL: https://codalab.lri.fr/competitions/204
Github repo of the project:
https://github.com/codalab/competition-examples/tree/master/basic-competition-bundles/Iris

[TEXT PAGE, WORTH 5 POINTS]

Background and motivation

We decided to work on the well-known Iris dataset from Fisher's classic paper [1]. This is a simple classification problem that attracted us because it will allow us to learn about machine learning problem in the context of a real world application in botany: distinguishing between 3 types of irises (virginica, setosa, and versicolor).

Data and problem description:

The problem is a classification problem. The data set contains 3 classes referring to types of iris plant.
One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.
The statistics of the dataset are summarized in **Table 1** (worth 2 bonus points).
More details are found on the README.ipynb file on our Github repository.

Approach chosen:

We will apply various methods of dimensionality reduction such as feature selection and PCA to bring the problem back to a projection in a 2 dimensional space. We will then apply various methods of classification found in scikit-learn [2] and perform a search for the best model/hyper-parameters using cross-validation (Figure 1). We use the hyper-parameter structure of AutoSKLearn [3]. One key difficulty is to perform an efficient search to keep our computational time within 10 minutes. We will implement and experiment with Particle Swarm Optimization (PSO) [4] and possibly other methods.

Brief description of the classes:

Binôme 1 will work on data visualization, following the class outline zDataManager.py in our Github repo. We also provided an outline for the corresponding test class zDataManagerTest.py. Typical figures will include correlation matrix heatmaps (Figure 2) and scatter plots (Figure 3).
Binôme 2 will work on preprocessing, following the class outline zPreprocessor.py in our Github repo (tested by zPreprocessorTest.py). This class will implement feature selection methods such as univariate feature selection based on correlation (Figure 2). We will also experiment with PCA.
Binôme 3 will work on selecting the classifier and combining preprocessing and classification via scikit-learn pipelines (classes zClassifier.py and zClassifierTest.py). Our model-selection/hyper-parameter selection method will combine cross-validation (Figure 1) and PSO. The pseudo-code the PSO method that we will implement is found in Algorithm 1.

Preliminary results:

Our README.ipynb notebook shows preliminary results, which are summarized in Table 2.
The results show that the problem is relatively easy (high accuracy on test data). Even though the problem is non-linearly separable, non-linear methods do not necessarily give better results. This is due to the small training set and the risk of overfitting.

[FIGURE PAGE, WORTH 5 POINTS]

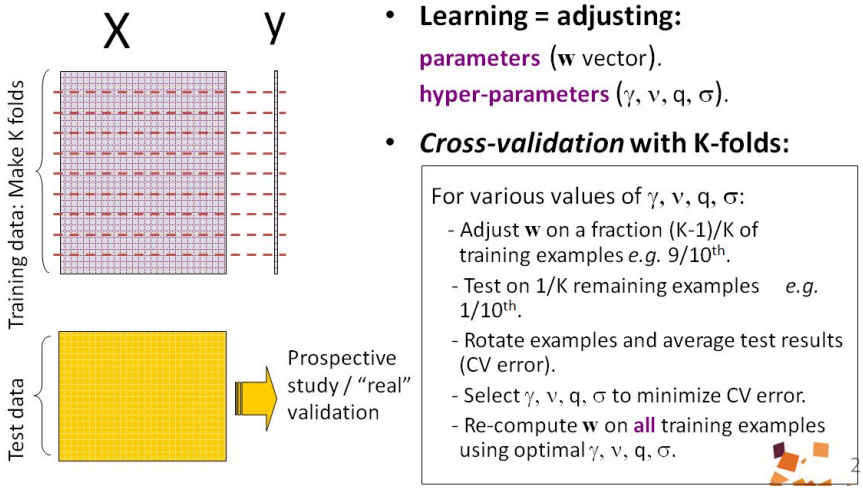

Figure 1: Cross-validation method. Class notes [6].

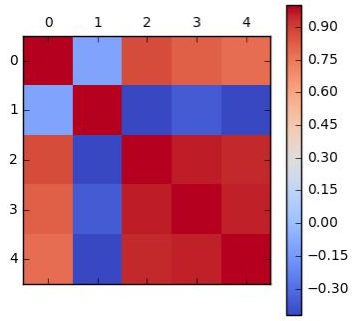

Figure 2: Correlation between features. The first 4 columns represent the features (sepal length and width). The last column represents the target (class values).

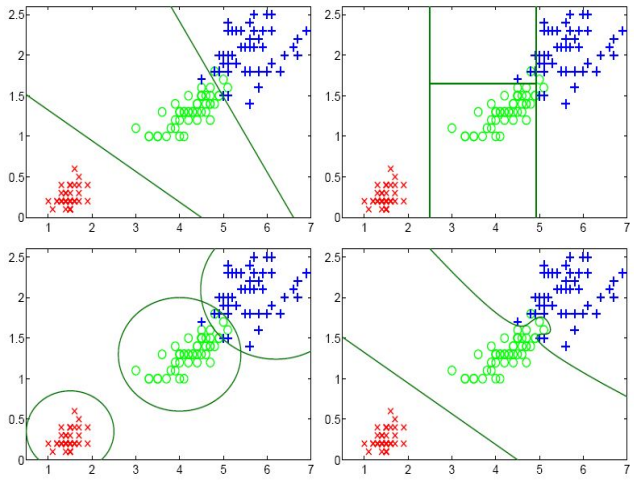

Figure 3: Two-dimensional representations of the Iris data. The colors indicate the classes (red=setosa, green=versicolor, blue=virginica) and the lines show the decision boundaries for 4 different classifiers (from top left to bottom right: linear, random forest, Gaussian, and SVM). Reference [5].

[BONUS PAGE (OPTIONAL), WORTH 5 POINTS]

Table 1: Statistics of the data (2 points).

Dataset	Num. Examples	Num. Variables/features	Sparsity	Has categorical variables?	Has missing data?	Num. examples in each class
Training						
Valid(ation)						
Test						

Table 2: Preliminary results (3 points).
Try 5 methods, some examples are given, but you may choose others.
Training refers to the training error. CV refers to the cross-validation error. Valid(ation) refers to the score obtained on the leaderboard when you make a submission to Codalab.

Method	NaiveBayes or Gaussian classifier	Linear regression or SVM	Decision Tree	Random Forest	Nearest Neighbors
Training					
CV					
Valid(ation)					

[ALGORITHM AND REFERENCE PAGE, WORTH 5 POINTS]

Algorithm 1 Particle swarm optimization (PSO).

Requires:
- $\mathbf{c}_1, \mathbf{c}_2$: weights for local and global information;
- $\mathbf{m}$: number of particles in the swarm;
- t_{max}: number of iterations;
- $\mathbf{W}$: inertia weight

Initialize swarm ($\mathbf{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$)
Compute fitness function $fitness(\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\})$
Identify global best ($\mathbf{g}^t$) solution
Identify personal best solutions ($\mathbf{p}_{1,\ldots,m} = \mathbf{x}_{1,\ldots,m}$)
$t = 1$
while $t < t_{max}$ **do**
  for all $\mathbf{x}_i \in \mathbf{S}$ **do**
    Calculate velocity $\mathbf{v}_i$ for $\mathbf{x}_i$ (Equation (2))
    Update position of $\mathbf{x}_i$ (Equation (1))
    Compute $fitness(\mathbf{x}_i)$
    Update $\mathbf{p}_i$ (*if needed*)
  end for
  Update $\mathbf{p}_g^t$ (*if needed*)
  Decrease $\mathbf{W}$
  t^{++}
end while
return $\mathbf{p}_g^t$

Algorithm 1: Pseudo-code of PSO. Reference [4].

References:

[1] The use of multiple measurements in taxonomic problems. R. A. Fisher. Annual Eugenics, 7, Part II, 179-188 (1936).

[2] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[3] AutoSKLearn: Efficient and Robust Automated Machine Learning, Feurer et al., Advances in Neural Information Processing Systems 28 (NIPS 2015).

[4] PSMO: Ensemble particle swarm model selection. Hugo Jair Escalante, Manuel Montes-y-Gómez, Luis Enrique Sucar. IJCNN 2010: 1-8.

[5] Learning machines. N. Jankowski and K. Grabczewski, in Feature extraction: Foundations and applications. I. Guyon et al. Eds. Springer, 2006.

[6] Class notes L2 mini-projets. Isabelle Guyon, 2017.