Optimizing classifier parameters for insect datasets

Bruno Gomes Coelho, André Maletzke, Gustavo Batista

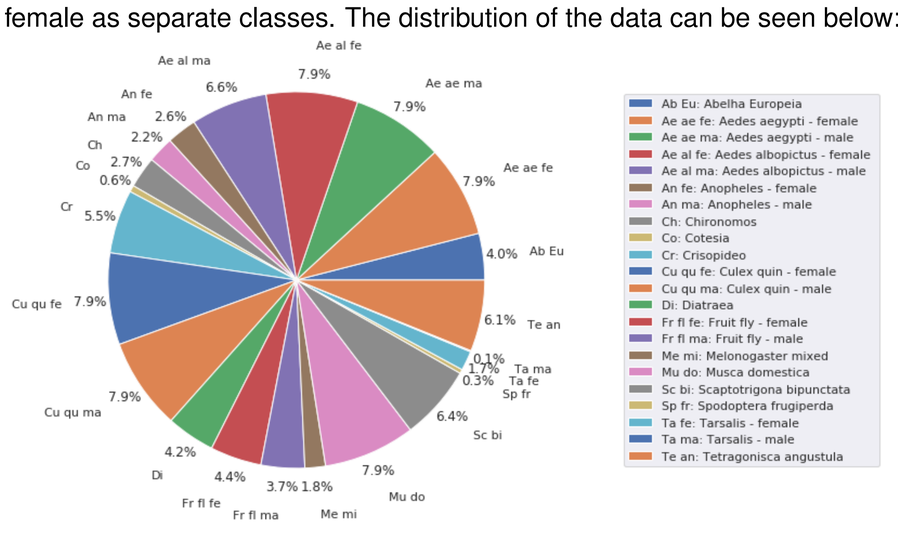
ICMC - Universidade de São Paulo
São Carlos, SP, Brasil
bruno.gomes.coelho@icmc.usp.br

1. Introduction and objectives

Motivated by the real life problem of being able to identify and then selectively capture dangerous insects that transmit various diseases, this poster analyzes the random search method of optimizing the parameters of two of the most recommended [1] machine learning algorithms, Support Vector Machines (SVM) and Random Forest (RF).

2. The problem

Utilizing an infrared light sensor designed by our research group for flying insects (Silva et al.), we acquired data on 22 classes of insects from 16 species, some considering male or female as separate classes. The distribution of the data can be seen below:



In total, we have 138,323 instances, each one with 41 features. The data was then divided into 3 independent problems, All Classes (AC), Mosquitoes versus non-mosquitoes (M) and Aedes aegypti female versus non-female (AF). For each one, 70% of the data was user for training while the remaining 30% was used for testing.
Using Python and the Scikit-learn library [5] classifiers were trained with the default parameters, generating SVM-D and RF-D. These will serve as our baselines.
For each classifier algorithm, there are many parameters that impact the accuracy. We chose two of the most important for each algorithm for tuning.

3. Support Vector Machines Parameters

For the SVM Optimized (SVM-O), we varied:

- The penalty of the error term, C, a real value from 0 to 30000.
- The kernel coefficient, KC, a real value from 0 to 10.

Only the Radial kernel was used since a complete search over the parameters encompasses the linear Kernel (Keerthi et al., 2003).

4. Random Forest Parameters

For the RF Optimized (RF-O) we varied:

- The number of trees in the forest, NT, a integer value from 10 to 500.
- The number of features, NF, to consider when looking for the best split, from 1 to 41.

5. Random Search

To vary the parameters mentioned above a random grid search using 5-fold cross-validation of the training data was used to check 60 random values (30 for each parameter) for each classifier. For all parameters, the random values were sampled from a uniform distribution given their range. The amount of values to check was calculated considering the probability of falling into the 5% interval of the true maximum of the given search space with a 95% probability (Bergstra et al.).
More specifically, let's consider *n* random values to be tested. Let's suppose that our search space encloses at least 5% of the close-to-optimal region of hyper-parameters, a reasonable consideration since we are analyzing big ranges.
Each random parameter has a 5% chance of landing in the top 5% interval. When all the values are drawn independently, then the probability that none of them fell into the interval is $(1 - 0.05)^n$
We are interested in the probability that at least one of our values fell into the interval, so we need to consider one minus our previous result. Since we are considering a 95% chance of success, we need a *n* such that:

$$1 - (1 - 0.05)^n > 0.95 \quad (1)$$

Solving this, we find $n >= 60$, which justifies our choice of using $n = 60$.

6. Results

The best parameters found were then used for training on the whole training set and evaluated on the test set. The SVM-O and RF-O had the following parameters for the datasets AC, M, and AF, respectively: (C:762, KC:1.0789), (C:762, KC:1.0789) and (C:11660, KC:2.7134), while the RF-O had (NT:215, NF:35), (NT:227, NF:18), and (NT:376, NF:28). The accuracies obtained were:

	SVM		RF	
	D	O	D	O
AC	93.82%	**97.07%**	95.87%	**97.31%**
M	97,82%	**99.01%**	98.94%	**99.07%**
AF	88.52%	**91.57%**	92.70%	**95.99%**

7. Conclusions

For both classifiers, the random search can be seen to give better results than the default parameters.

References

[1] Wu, X., et al. (2008). *Top 10 algorithms in data mining*. Knowledge and information systems, 14(1), 1-37.

[2] Bergstra, J., Bengio, Y. (2012). *Random search for hyper-parameter optimization*. Journal of Machine Learning Research, 13(Feb), 281-305.

[3] Silva, D.F., et al. (2011). *Resultados Preliminares na Classificação de Insetos Utilizando Sensores Óticos*.

[4] Keerthi, S.S., Lin, C.J. (2003). *Asymptotic behaviors of support vector machines with Gaussian kernel*. Neural computation, 15(7), 1667-1689.

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

Financing