Report
"Advanced Learning Models"

PHAM Tuan Hiep
master MOSIG Data Science
tuan-hiep.pham@grenoble-inp.org

PHAN Ly Huynh
master MOSIG Data Science
ly-huynh.phan@grenoble-inp.fr

1 The first approach.

In this approach, we used spectrum kernel and bag of words technique for data preprocessing (4, 5 grams used). After this step, we found out the algorithms to classify the data, we test Perceptron, Soft Support Machine Vector, Logistic Regression. We also define a simple function to compute the accuracy score and test with each algorithm. We found that logistic regression has the best performance in these algorithms in our experiment. So, we choose it. The activation function used is sigmoid function:

$$S(x) = \frac{1}{1 + e^{-x}}$$

Besides, to avoid overfitting problem, we use l_2 regularization:

$$R(\omega) = \lambda ||\omega||_2^2$$

To improve the accuracy score, we split the train dataset into train data and test data set by the ratio 2:1 and then train our model with different learning rate η and regularization parameter λ. and then run it to find out the best parameters which are evaluated by the accuracy score.

2 The second approach

This is a simple approach based on probabilistic. With each ADN sequential, we use n-grams to encode data to features $X_1, X_2, ..., X_k$. We assume that each feature has its probability to decide the state of general sequence. For example, $p(y|X_i)$ is the probability that a sequential having X_i has state y and the probability of a sequence having features $X_1, X_2, ..., X_n$ has the state y is:

$$p(y|X_1, X_2, ..., X_n) = \prod_{i=1}^{n} p(y|X_i)$$

$$p(y|X_i) = \frac{p(y, X_i)}{p(X_i)}$$

To avoid the case $p(X_i) = 0$ or $p(y, X_i) = 0$ affects much the result, we use the simple smooth method:

$$p(y|X_i) = \frac{p(y, X_i) + 1}{p(X_i) + 1}$$

However, the sequences of ADN, which have different orders of features usually have different states. Thus, to improve the accuracy, we use also $p(X_i|X_{i-1})$ to estimate the probability of se-

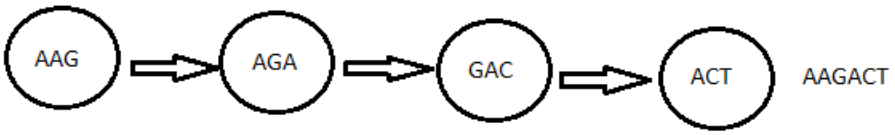
quence. So, the general formula is:

$$p(y|X_1X_2...X_n) = \prod_{i=1}^{n} p(y|X_i)p(X_i|X_{i-1})_{state=y}$$

$$p(y|X_i) = \frac{p(y, X_i) + 1}{p(X_i) + 1}$$

$$p(X_i|X_{i-1})_{state=y} = \frac{p(X_iX_{i-1}, y)}{p(X_{i-1}, y)}$$

With each sequence, we estimate $p(1|X_1X_2...X_n)$ and $p(0|X_1X_2...X_n)$ and decide the label of sequence following the higher probability.
Example, with sequence S=**AAGACT**, n-grams = 3, we have the model, $X_1 = AAG, X_2 = AGA, X_3 = GAC, X_4 = ACT$.



$$p(y|S) = p(y|X_1) * p(X_2|X_1)_y * p(y|X_2) * p(X_3|X_2)_y * p(X_3) * p(X_4|X_3)_y * p(y|X_4)$$

With the data set of this project, we tried this model with n-grams = 4,5,6 and chose n-grams = 6.

3 Our project

Our project which contains both data and source code is available in github repo link. We are welcome all contribution.