Page blocks classification: Support vector machines, and neural networks in R

Nyasha C

Data Science – Supervised Learning 2019

DEPARTMENT OF STATISTICAL SCIENCE

University of Cape Town

The problem concerns classifying all blocks of a page layout of a document. The blocks have been detected using a segmentation process - an essential step in document processing in order to separate text from graphics. The five classes are:

- **Text**
- **Horizontal line**
- **Picture**
- **Vertical line**
- **Graphic**

The volumes, per class are given in the table below,

Table 1. Distribution of classes in sample data

Class	Description	Number of observations
1	Text	4420
2	Horizontal line	303
3	Picture	23
4	Vertical line	79
5	Graphic	100

It is quite evident that there is some class imbalance, almost 90% of observations belong to the text class, while picture and vertical line have the lowest sample volumes.
Two classification techniques are presented, namely Support Vector Machines, and Neural Networks. The training of these models and their relative performance is subsequently presented.

Support Vector Machine

The support vector machine (SVM) is an extension of the support vector classifier (SVC). The support vector classifier classifies an observation depending on which side of the hyper-plane it lies; the classifier allows for some misclassification and is ideal for problems where the classes are linearly separable. SVM's allow for the accommodation of non-linear boundaries between classes.

While the representation of the linear classifier can be done using inner products, SMV's generalize this representation by replacing the inner product with a kernel,K. The kernel quantifies the similarity of two observations.

The linear kernel is represented as follows,

$$K\left(x_i\ ,\ x_{i'}\right)\ =\ \sum_{j=1}^{p} x_{ij}x_{i'j}$$

and the similarity measure in this case is simply the Pearson correlation coefficient. The functional form of the linear kernel is given by:

$$f(x) = \beta_0 + \sum_{j=1}^{p} x_{ij}x_{i'j}$$

Non-linear kernels such as the polynomial, and radial kernel result in more appropriate decision boundaries when faced with non-linear data. Figure 1 below shows an example of linearly separable, and non-linearly separable data, using a linear kernel in the latter would result in poor performance.

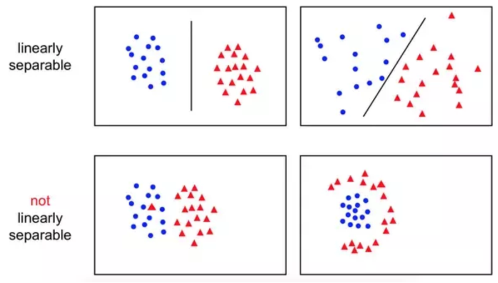

Fig. 1. Linear, and non-linearly separable classes. The choice of kernel is dependant on the separation of classes

The radial kernel,which is more widely used than the polynomical kernel takes the form of:

$$K\left(x_i\ ,\ x_{i'}\right)\ =\ \exp\left(-\gamma\sum_{j=1}^{p}\left(x_{ij}-x_{i'j}\right)^2\right)$$

The functional form of the SVM with a radial kernel is given by:

$$f(x) = \beta_0 + \exp\left(-\gamma\sum_{j=1}^{p}\left(x_{ij}-x_{i'j}\right)^2\right)$$

SVM Hyperparameters. While SVM's allow for the misclassification of observations, a cost, C is placed on these errors. The larger the cost, the lower the number of training errors. A cost that is too high will result in a model that performs well on the training set but is likely to over-fit on the test data set. For a linear kernel or simply the support vector classifier, the only hyper-parameter we need to specify is the cost.

For the radial kernel we also need to specify an appropriate γ, here γ is a positive constant. The radial kernel works by computing the Euclidean distance between a given observation x^* and each training observation x_i. For large distances the value of $\exp\left(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i\prime j})^2\right)$ will be small, and those observations will have little influence on $f(x^*)$.As γ increases the influence of x_i will decrease.

SVM Parameter tuning. Both parameters are tuned using the tune.svm function in the R package e0171.This package allows us to use cross-validation to find the most appropriate hyper-parameters. The default number of folds is 10, for our data set this will leave sufficient volumes in each fold so this remains unchanged.

For the radial kernel, we must first define an appropriate space for tuning c.We begin by setting gamma = 1 and investigate the impact of various values of c on the test misclassification error rate. We will then search over a grid of combined c and γ values. The model with the lowest test error will be selected as our final model, together with the corresponding c and γ.
The data set was split into a training set containing 80% of the observations and a test set containing 20% of the observations. The data is scaled before training the model.
We test values between 1 and 20 increasing the cost by 0.2 each time. The results indicate that increasing the cost for constant values of γ does not provide significantly different results, especially for cost values that are close together. For the final grid search we search over a smaller space, 1 through 10 in increments of 1.
The best model, selected through 10-fold cross-validation has the following parameters:

$$\gamma = 0.1$$

$$c = 9$$

The accuracy on the training, and test data sets is presented below:

Table 2. Misclassification error rate for radial kernel withγ=0.1 c=9

Sample	Sample size	Misclassification Error Rate
Training	3940	2.31%
Test	985	3.35%

Despite the imbalance of classes, the model performs well even on the low-volume classes. In table 3 below, the distribution of true vs. predicted classes is given below.

Table 3. Confusion matrix for SVM with radial kernel, cost = 9 and $/gamma$ = 0.1

	1	2	3	4	5
1	3513	23	0	10	32
2	16	210	0	1	1
3	1	0	19	0	0
4	1	1	0	56	0
5	4	1	0	0	51

A linear kernel, or support vector classifier is also trained. For the linear kernel we only specify the cost hyper-parameter and optimise it using 10-fold cross-validation. The best model, selected through cross-validation has a cost = 6.The accuracy on the training, and test data sets is presented in table 4. Based on the cross-validation error we would choose the radial kernel with c = 9 and γ=0.1. The models perform similarly on the test data set. In the next section we train a neural network on the blocks data set.

Table 4. Misclassification error rate for linear kernel with c=6.

Sample	Sample size	Misclassification Error Rate
Training	3940	3.47%
Test	985	3.35%

Neural Networks

Neural networks have well documented success in computer-vision, speech detection and natural-language processing. The Artificial Neural Network (ANN) models the relationship between a set of input signals (predictors), and an output signal (the dependent variable).
Neural networks can be can be defined by two broad categories, Feed-forward networks, and Recurrent/feedback networks. An illustration of a Feed-forward NN is given in the image below figure 2 below.

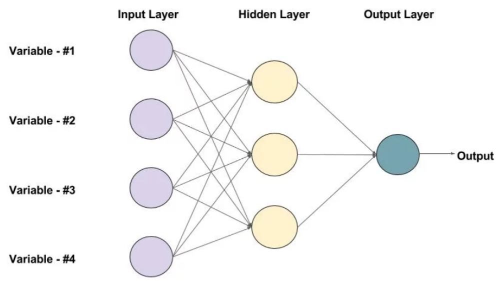

Fig. 2. A simple feed-forward neural network A simple feed-forward neural network with for predictors,and one hidden layer with 3 hidden units., and three

The neural network, like the human brain uses a network of interconnected cells (artificial neurons in the case of ANN) in order to solve learning problems. In the case of the ANN, each neuron in the hidden layer is connected to the input layer via a vector of parameters or weights. These neurons are processed with a nonlinear function. The output layer, like the input layer also has weights. The output layer is simply a transformation of the combined inputs. An activation function is used for the transformation.

Neural Network characteristics. The various neural networks can be compared according to the following framework:

- Activation function - usually sigmoid, or tanh.
- Network topology - number of neurons, and hidden layers.
- The training algorithm.
- Loss function - cross-entropy in the case pf a many-class classification problem.

.1. Activation function. As mentioned above, the activation function serves as a processor for incoming information. The incoming information passes through the activation function, and into the network.
The two most common activation functions are the sigmoid and the hyperbolic tangent function (tanh). The sigmoid activation function results in a continuous output, between 0 and 1(inclusive). The function is given below:

$$f(Z) = \frac{1}{1+\exp(-\alpha z)}$$

Tanh produces a continuous output between -1 and 1(inclusive) and takes the form:

$$f(Z) = \frac{\exp(Z)-\exp(-Z)}{\exp(Z)+\exp(-Z)}$$

.2. Network topology. Network topology refers mainly to the number of neurons, the number of layers, as well as the direction in which information travels in the network. This structure is the basis of the ability of the network to learn. A more complex topology generally lends itself to more complex problems - problems with subtle patterns, and complex decision boundaries.
Hyper-parameter tuning is used to find the optimal topology - specifically the number of hidden layers, and hidden units.There is no rule-of-thumb for finding the optimal number of hidden layers. One or two hidden layers are usually sufficient. Although large, complex neural networks may sound appealing for capturing complexities in data, they tend to over-fit, and are computationally expensive.

.3. Training methods. Training, or learning methods are concerned with finding the weights, and biases in the network. The weights can either be set explicitly using prior knowledge, or by allowing the neural network to change the weights through learning rule. The latter is known as back-propagation, this includes a forward phase,and a backward phase. With the backward phase comparing the output signal to the true value, and adjusting the network weights accordingly. One cycle of this algorithm is known as an epoch. The following algorithm is repeated until convergence is reached:

$$w_j := w_j - \alpha \frac{\delta}{\delta w_j} J(w_0, ..., w_j)$$

The algorithm makes decision based on the following question 'What is the slope of J at point (x_j) and by how much will the error change for a change in the weight'. α specifies the size of the steps that must be taken to reach the optimal values, if α is too large there is a risk that the algorithm may never converge because it will keep missing the optimal value.

Deep Neural Network

Deep neural networks(DNN) refer to networks with a network topology consisting of more than two layers (one hidden, and one output). These complex models are well-known for their applications in image-recognition, and self-driving cars, and will thus be suitable for our page block classification problem.
The activation function typically used for DNN is the rectifier which is defined as follows:

$$f(Z) = \max(0, Z)$$

where f(Z) is 0 when Z is less than Z, and Z elsewhere.

A. Training methodology for DNN. The following steps are taken in the training of the DNN:

- Data normalisation
- Weight initialisation
- Define number of epochs
- Learning rate, α
- Regularisation - L1 -L2 penalises large networks,as well as dropout which randomly sets nodes in the network to zero and encourages generalisation(less over-fitting).

B. Training the DNN on page blocks data. The h2o package is used for model training. We use grid search to tune the regularisation hyper-parameters, as well as the activation function. The default hidden layer of 2 with 20 hidden units is used. The models are ranked on their cross-validation accuracy, and the model with the highest accuracy is chosen. The results of the top 4 models, ranked on accuracy are presented in the table below.

Table 5. Training error rates for DNN

Activation	L1	L2	Cross-validation accuracy
Maxout	0.001	0.001	96.85%
Maxout	1.0E-4	0.0	96.75%
Maxout	0.001	1.0E-4	96.65%
Rectifier	1.0E-5	0.001	96.65%

The confusion matrix for the best model is given in table 6 below.

Table 6. Confusion matrix for DNN with maxout activation function

	1	2	3	4	5	Error	Rate
1	3515.00	14.00	1.00	1.00	4.00	0.01	20 / 3,535
2	35.00	197.00	0.00	1.00	2.00	0.16	38 / 235
3	10.00	0.00	9.00	0.00	0.00	0.53	10 / 19
4	10.00	1.00	0.00	54.00	2.00	0.19	13 / 67
5	52.00	0.00	0.00	0.00	32.00	0.62	52 / 84
Totals	3622.00	212.00	10.00	56.00	40.00	0.03	133 / 3,940

It is quite evident that despite the high overall accuracy, the model does not predict well on classes 2-4.

Conclusion

The overall results for the top 3 models are presented in the table 7. Based on the cross-validation results (training data) we would choose the SVM with the radial kernel because it has the lowest misclassification error rate on this sample. The test set indicates the the models tend to perform similarly on unseen data, although we saw in the confusion matrix that the accuracy for the DNN is driven by the accuracy of the classification of the most voluminous class, the text class. The SVM with the radial kernel is thus applied on the unseen test data set.

Table 7. Final model comparison

Model	Cross-validation error	Test error
DNN (Maxout)	3.37%	3.15%
SVM (Radial kernel)	2.31%	3.35%
SVM (Linear kernel)	3.47%	3.35%

References

Linear and non-linearly separable classes: C19 Machine learning A.Ziesserman http://www.robots.ox.ac.uk/ az/lectures/ml/lect2.pdf

A simple feed-forward neural network: Understanding Feedforward Neural Networks V.Gupta https://www.learnopencv.com/understanding-feedforward-neural-networks/

“Introduction to Statistical Learning” by Gareth James, Daniela Witte, Trevor Hastie and Robert Tibshirani

“The Elements of Statistical Learning” by Trevor Hastie, Robert Tibshirani and Jerome Friedman
Lecture Slides Supervised Learning by Sebnem Er (University of Cape Town)