UDK 621. 391

MATHEMATICAL CONCEPTS OVERVIEW OF CLUSTER ANALYSIS, SVM AND NEURAL NETWORKS IN MASS SERVICE NETWORKS

Kuleshova V.V., Lozhkovskyi A.G.

O.S. Popov Odessa national academy of telecommunications,
1 Kuznechna St., Odessa, 65029, Ukraine.
aloshk@onat.edu.ua ,vkuleshova7@gmail.com

ОГЛЯД МАТЕМАТИЧНИХ КОНЦЕПТІВ КЛАСТЕРНОГО АНАЛІЗУ, SVM ТА НЕЙРОННИХ МЕРЕЖ В МЕРЕЖАХ МАСОВОГО ОБСЛУГОВУВАННЯ

Кулешова В.В., Ложковський А.Г.

Одеська національна академія зв'язку ім. О.С. Попова,
65029, Україна, м. Одеса, вул. Кузнечна, 1.
aloshk@onat.edu.ua ,vkuleshova7@gmail.com

ОБЗОР МАТЕМАТИЧЕСКИХ КОНЦЕПТОВ КЛАСТЕРНОГО АНАЛИЗА, SVM И НЕЙРОННЫХ СЕТЕЙ В СЕТЯХ МАССОВОГО ОБСЛУЖИВАНИЯ

Кулешова В.В., Ложковский А.Г.

Одесская национальная академия связи им. А.С. Попова,
65029, Украина, г. Одесса, ул. Кузнечная, 1.
aloshk@onat.edu.ua ,vkuleshova7@gmail.com

Abstract. Nowadays, mass service networks are researched by using a variety of methods. However, the results that have been achieved this far are intermediate ones, especially, in case of multi-service telecommunication networks. The present paper provides an overview of cluster analysis, SVM and neural networks algorithms. The comparative analysis of mentioned models' prediction results was conducted and it was practically showed that that neural networks can provide a powerful tool for the telecommunications industry because the Neural Networks are nonlinear dynamic systems able to offer a solution almost anywhere where classic methods have failed. Neural Networks have certainly become one of the key technologies nowadays, though a critical analysis reveals some drawbacks.

Key words: neural networks, training algorithm, neuron, cluster analysis, support vector machine.

Анотація. Сьогодні мережі масового обслуговування досліджуються за допомогою різних методів. Проте результати, які були досягнуті до теперішнього часу, є проміжними, особливо у випадку мультисервісних телекомунікаційних мереж. Даний документ містить огляд кластерного аналізу, алгоритмів SVM та нейронних мереж. Проведено порівняльний аналіз результатів прогнозування згаданих моделей та практично показано, що нейронні мережі можуть стати потужним інструментом для телекомунікаційної галузі, оскільки нейронні мережі – це нелінійні динамічні системи, здатні запропонувати рішення майже в будь-якій ситуації, де не справляються класичні методи. Нейронні мережі, безумовно, стали однією з ключових технологій в наші дні, хоча критичний аналіз показує низку недоліків.

Ключові слова: нейронні мережі, алгоритм навчання, нейрон, кластерний аналіз, SVM.

Аннотация. В настоящее время сети массового обслуживания исследуются с использованием самых разных методов. Однако достигнутые до сих пор результаты являются промежуточными, особенно в случае мультисервисных телекоммуникационных сетей. В настоящем документе представлен обзор кластерного анализа, алгоритмов SVM и нейронных сетей. Проведен сравнительный анализ результатов прогнозирования упомянутых моделей и практически продемонстрировано, что нейронные сети могут стать мощным инструментом для

телекоммуникационной отрасли, поскольку нейронные сети представляют собой нелинейные динамические системы, способные предложить решение практически в любых случаях, где классические методы не справляются. Нейронные сети, безусловно, сегодня стали одной из ключевых технологий, хотя критический анализ выявляет некоторые недостатки.

Ключевые слова: нейронные сети, алгоритм обучения, нейрон, кластерный анализ, SVM.

There are many scheduling methods presented in literature, most of them based on Data science algorithms, multilayer supervised networks and other competitive algorithms. All these models are capable to deal with bursty traffic in any conditions.

Let us consider cluster analysis for data classification and compare support vector machine algorithm with neural networks concepts for analytics and predictive tasks:

1. Cluster Analysis - is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those ones in other groups (clusters).

Let us consider k-mean method:

```
import numpy as np
import matplotlib.pyplot as plt from
sklearn.cluster import KMeans %matplotlib
inline
```

As an example, we generated data for the three different segments in the form of the records shown below:

(age, gender, category-1, category-2, category-3)

```
c1 = np.array([19., -0.6, 100., 44., 5.])
c2 = np.array([26., -0.8, 110., 55., 200.])
c3 = np.array([45., -0.77, 90., 70., 20.])
v1 = np.array([5., 0.15, 20., 10., 6.])
v2 = np.array([7., 0.1, 30., 23., 20.])
v3 = np.array([12., 0.1, 32., 27., 12.])

d1 = np.random.normal(c1, v1, size=(200,5))
d2 = np.random.normal(loc=c2, scale=v2, size=(120,5))
d3 = np.random.normal(loc=c3, scale=v3, size=(150,5))
data = np.vstack((d1,d2,d3))

## Let us do some clean-up after data generation for i in
range(data.shape[0]):
    a,g,s1,s2,s3 = data[i] if a
    < 10: a = 10.
    if g < -1.: g = -1.
    if g > 1.: g = 1.
    if s1 < 0.: s1 = 0.
    if s2 < 0.: s2 = 0.
    if s3 < 0.: s3 = 0.
    data[i] = np.array([a,g,s1,s2,s3]) d_min =
np.min(data, axis=0)
```

```
d_max = np.max(data, axis=0)
data = (data - d_min) / (d_max - d_min)
```

Now the data is normalized, which means all vectors are distributed inside the unit cube. The example of data normalization:

```
plt.scatter(data[:,1:2], data[:,2:3])
plt.axis('equal')
plt.plot([0,0,1,1,0], [0,1,1,0,0], 'r')
plt.show()
```

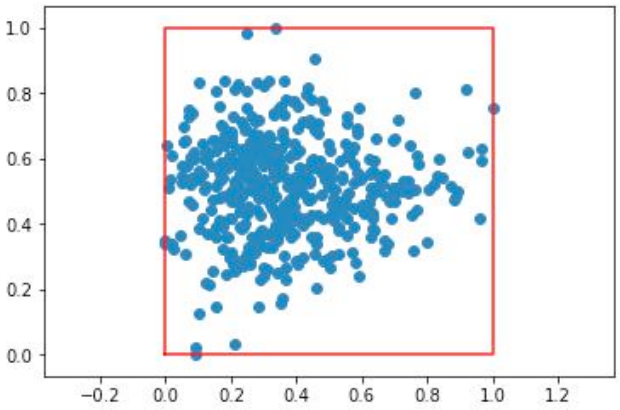

Figure 1 – Distributed normalized data

Customer age distribution:

```
In [219]:
plt.hist(data[:,0:1]*(d_max-d_min)[0] + d_min[0], bins=20)

plt.show()
```

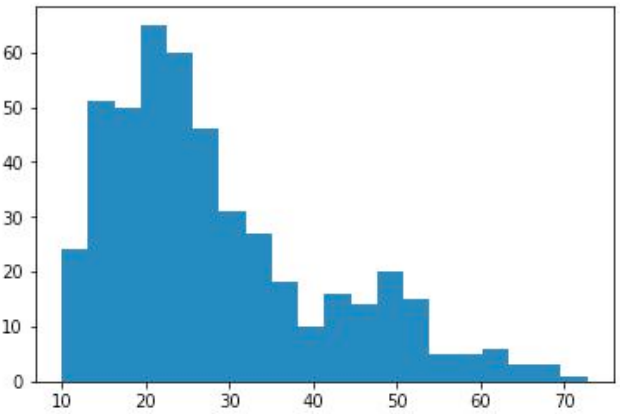

Figure 2 – Custom age distribution

Let us look at how category 1 is distributed vs. category 2:

```
fig = plt.figure(figsize=(7,7))
plt.axis('equal')
plt.scatter(x=d1[:,2:3], y=d1[:,3:4],c='r')
plt.scatter(x=d2[:,2:3], y=d2[:,3:4],c='b')
plt.scatter(x=d3[:,2:3], y=d3[:,3:4],c='g')
plt.show()
```

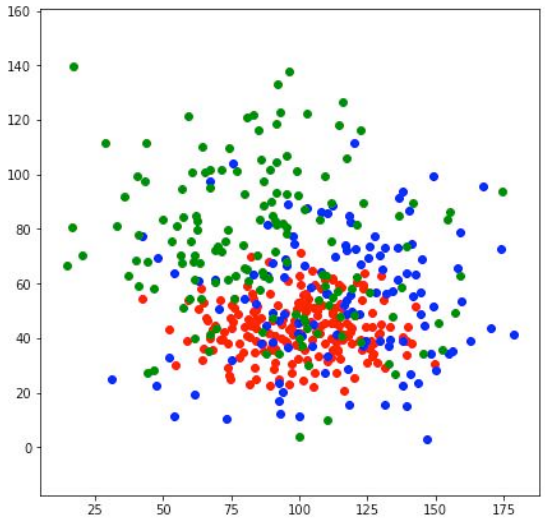

Figure 3 – Distribution by categories

2. Support Vector Machine - supervised ML algorithm for data classification

Training dataset:
Let us consider 5 labeled samples for each class:

```
from sklearn import svm
X1 = d1[:20]
Y1 = np.ones(20).astype('int')

X2 = d2[:20]
Y2 = 2 * np.ones(20).astype('int')

X3 = d3[:20]
Y3 = 3 * np.ones(20).astype('int')
X = np.vstack((X1,X2,X3))
Y = np.hstack((Y1,Y2,Y3))

#clf = svm.SVC(kernel='linear')
clf = svm.SVC()
clf.fit(X, Y)
print "Correctly predicted 1: %i%%" % (100*clf.score(d1,
np.ones(d1.shape[0]).astype('int')))
print "Correctly predicted 2: %i%%" % (100*clf.score(d2, 2*np.ones(d
2.shape[0]).astype('int')))
print "Correctly predicted 3: %i%%" % (100*clf.score(d3, 3*np.ones(d
3.shape[0]).astype('int')))
```

We have got the predicted results:

```
Correctly predicted 1: 15%

Correctly predicted 2: 100%

Correctly predicted 3: 14%
```

3. Neural Networks

The fundamental idea of artificial neural is to assemble several single simple processors that interact through a dense web of interconnections, which result in a network architecture that is unlike the sequential linear processing and architecture of conventional computer systems. A neural network consists of a large number of simple processing elements called neurons or nodes. Each neuron is connected to other neurons by means of directed communication links, each one with an associated weight. The weights represent information being used by the network to solve a problem.

One of the most popular neural network architecture is the multi-layer feed forward neural network (MLNN), also called multi-layer perceptron (MLP :

1. Each processing element (essentially a neuron) receives inputs from other elements.
2. The inputs are weighted and added.
3. The result is then transformed (by a transfer function) into the output.

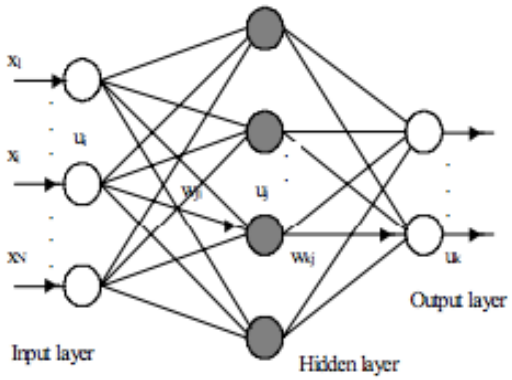

Figure 5 – An example of a multilayer neural network

The most common types:

- Feedforward Networks
- Recurrent Neural Networks (RNN)
- Convolutional Neural Networks (CNN)

$\underset{a_0}{\rightarrow}$ - input layer

$\underset{a_i}{\rightarrow}$ - output from i-th layer (1)

$$W_i \cdot \underset{a_i}{\rightarrow} = \underset{z_i+1}{\longrightarrow}$$

$$\sigma\left(\underset{z_i+}{\longrightarrow}\right) = \underset{a_i+1}{\longrightarrow} \quad (2)$$

Training Data:

List of pairs: $[(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)]$

$$F(x, W) \rightarrow \hat{y},\ \sigma = y - \hat{y}$$

$$W \leftarrow back - propagation\ (W, \sigma) \quad (3)$$

Stochastic Gradient Descent

Batches and Epochs

```
   - batch 1    \
   - batch 2     \
 -   batch 3      )    epoch
 - ...           /
 -   batch k    /
```

In [230]:

```
from sklearn.neural_network import MLPClassifier

clf = MLPClassifier(solver='lbfgs', alpha=1e-5,hidden_layer_sizes= (5, 3),
random_state=1)
X1 = d1[:20]

Y1 = np.ones(20).astype('int')

X2 = d2[:20]

Y2 = 2 * np.ones(20).astype('int')

X3 = d3[:20]

Y3 = 3 * np.ones(20).astype('int')

X = np.vstack((X1,X2,X3))

Y = np.hstack((Y1,Y2,Y3))
```

In [231]:

```
clf.fit(X, Y)
```

Out[231]:

```
MLPClassifier(activation='relu', alpha=1e-05, batch_si
ze='auto', beta_1=0.9,
        beta_2=0.999, early_stopping=False, epsilon=1e-

08,

        hidden_layer_sizes=(5, 3), learning_rate='const

ant',

        learning_rate_init=0.001, max_iter=200, momentu

m=0.9,

        nesterovs_momentum=True, power_t=0.5, random_st ate=1,
shuffle=True,
        solver='lbfgs', tol=0.0001, validation_fraction =0.1,
verbose=False,
        warm_start=False)
```

In [232]:

```
print "Correctly predicted for 1: %i%%" %(100. * sum( 1 for p in
clf.predict(d1) if p==1) / d1.shape[0])

print "Correctly predicted for 2: %i%%" %(100. * sum( 1 for p in
clf.predict(d2) if p==2) / d2.shape[0])
print "Correctly predicted for 3: %i%%" %(100. * sum( 1 for p in
clf.predict(d3) if p==3) / d3.shape[0])
```

We have got the predicted results:

```
Correctly predicted for 1: 84%

Correctly predicted for 2: 90%

Correctly predicted for 3: 32%
```

The results of the research show that the differences in prediction between neural networks and SVM are statistically significant.

Conclusion: The present paper provides the analysis of mathematical concepts algorithms that show that the cluster analysis k-means gives the proper results for classification problems, and neural networks give the higher predicted accuracy than the SVM, which allows determining mass service networks' characteristics. Building and teaching a neural network model is a complex process due to the incoming data being difficult to convert. Neural networks can be used to solve lots of problems in telecommunications, such as coding, decoding and error correcting codes, image processing, fraud detection, software analysis, predictive modelling. Neural networks can be used for mass service networks design, management, routing and control, which have thousands of nodes, deal with very different traffic types and deserve a huge number of users. Most of the QoS parameters are variable in time. The new services in the modern networks require dynamic channel assignment, interference avoidance, propagation prediction and automated planning techniques. And still the problems are not solved with current algorithms and methods.

Neural networks are promising, elegant, intelligent solutions due to their capability to assure an adaptive, flexible, optimal control and an extraordinary processing speed, they are capable of learning and to predict some of the parameters.

Although neural networks have proved the capability to solve many difficult problems, a critical analysis reveals some disadvantages. In spite of that, a lot of successful products have already showed that neural networks can provide a powerful tool for telecommunications industry.

REFERENCES:

1. Edwards T, Tansley D.S.W, Frank R.J. & Davey N. Traffic Trends Analysis Using Neural Networks. // Proceedings International Workshop on Applications of Neural Networks to Telecommunications 3 (IWANNT'3), 2003. – P.157–164.
2. Field, S.D.H., Davey, N., Frank, R. J./Using Neural Networks to Analyse Software Complexity// Australian Journal of Intelligent Information Processing Systems. – 1999. – Vol 3, No. 3. – P.14–32.
3. Corina Botoca, Georgeta Budura. Using Cellular Neural Network To Adaptive Equalization//Proceedings of the Symposium on Electronics and Telecommunications, ETC. – 2000. – P. 245– 49.
4. Krylov V.V. Teorija teletrafika i ejo prilozhenija / V.V. Krylov, S.S. Samohvalova. – SPb.: BHV-Peterburg, 2005. – 288 p.
5. Lozhkovskyi A.G. Teoriya masovogo obslugovuvannya v telekomunIkatsIyah / A.G. Lozhkovskyi. – Odesa: ONAZ ім. O.S. Popov, 2012. – 112 p.

ЛИТЕРАТУРА:

1. Edwards T, Tansley D.S.W, Frank R.J. & Davey N. Traffic Trends Analysis Using Neural Networks. // Proceedings International Workshop on Applications of Neural Networks to Telecommunications 3 (IWANNT'3), 2003. – P.157– 64.
2. Field, S.D.H., Davey, N., Frank, R. J./Using Neural Networks to Analyse Software Complexity// Australian Journal of Intelligent Information Processing Systems.- 1999.-Vol 3, No. 3 – P.14–32.
3. Corina Botoca, Georgeta Budura. Using Cellular Neural Network To Adaptive Equalization//Proceedings of the Symposium on Electronics and Telecommunications, ETC. – 2000. – P. 245–249.
4. Крылов В.В., Самохвалова С.С. Теория телетрафика и её приложения / СПб.: БХВ-Петербург, 2005. – 288 с.
5. Ложковский А.Г. Теория массового обслуживания в телекоммуникациях / А.Г.Ложковский // Одесса: ОНАЗ им. О.С. Попова, 2012. – 112 с.