

# MENTAL STATE'S LONG/SHORT RUNNING PREDICTIONS. CLASSIFICATION OVER REGRESSION METHODS IN ADVANCE ANALYTICS

Zuev E. (Russian Federation)

*Zuev Egor Dmitrievich – Bachelor,  
DEPARTMENT COMPUTER SYSTEMS AND NETWORKS,  
HIGHER SCHOOL OF ECONOMICS, MOSCOW*

**Abstract:** *this paper outlines the way in which the aggregated data from EEG device may be used in future predictions; it dives into the problems, related to building a regression function, based on certain conditions of certain user; classify certain function by timestamp; and suggest its implementation. Also, the following article covers the basic idea of classification in terms of taking a decision, which function to use, in order to calculate correct average concentration level of current user on certain time frame.*

**Keywords:** *linear regression, Naïve Bayes, SVM, automatization, EEG., brain waves, Javascript, fuzzy logic, machine learning, classification.*

## КОРОТКОСРОЧНЫЕ/ДОЛГОСРОЧНЫЕ ПРЕДСКАЗАНИЯ МЕНТАЛЬНОГО СОСТОЯНИЯ.

### КЛАССИФИКАЦИЯ РЕГРЕССИОННЫХ МЕТОДОВ В ПРОДВИНУТОЙ АНАЛИТИКЕ

Зуев Е. Д. (Российская Федерация)

*Зуев Егор Дмитриевич – бакалавр,  
кафедра компьютерных систем и сетей,  
Высшая школа экономики, г. Москва*

**Аннотация:** *в данной статье рассматривается возможность использования собранных данных с ЭЭГ устройства для будущих предсказаний; Также будут рассмотрены проблемы построения регрессионного уравнения для конкретного пользователя; классификация определенной функции на основе временного промежутка и будет предложена собственная реализация решения данной проблемы. Более того, в данной рукописи будет рассмотрена базовая идея применения методов классификации для определения, по какой функции считать средний уровень концентрации конкретного пользователя для данного временного фрейма.*

**Ключевые слова:** *линейная регрессия, наивный Байес, SVM, ЭЭГ, мозговые волны, Javascript, ленивые вычисления, машинное обучение, классификация.*

#### 1. INTRODUCTION

Time passes by..., from Turing machine up to modern high scale clusters, humanity made a great breakthrough for the past 50 years. From the very beginning, the main aim of information technologies was collecting and passing information between nodes. But when the aim was achieved, a new one field appeared – telemedicine. This new flow serves as a solution for one of the most problems, which everyone faces with – it's a health tracking. When we have enough information about certain diseases, we can track it by ourselves, at least, on first steps – as in many cases, it can be detected by relative symptoms.

The suggested solution is aimed to track user's mental states through BCI interface and predict his future state. As an example, for the proof of the concept – our subject will be tracking the exact state – concentration state, and see how it will be changed over time. The collected data will allow us to predict user's future mental state, and detect mental problems, which may occur in, for instance, stress situations.

The rest of the paper is organized as follows: Section 2 gives a shortcut observe of problems, related to predictions for certain user, by using pure aggregated data. Section 3 describes a way of choosing function for calculation concentration level on certain time stamp.

#### 2. ONE NODE PREDICTION

By a «one node prediction», we are talking about a mechanism, which predicts mental state by pure data (which has been just collected). The main disadvantage of this approach is in that function, by which the prediction will be made, as it's not rely on certain user. But for general purpose, this approach could be used to calculate the average

data (in our case – concentration level). So, the first thing which come up to mind is to introduce a special coefficient, which will make the calculation smoother. However, when the system is looking more like a blackbox, this flow will not work. From this moment, let's assume, that we won't ever get enough good result, which will satisfy us, if we will run calculations in isolatable way. For this purpose, we will keep in mind previous aggregated and calculated data, in order to apply this coef to new function. This conception represents a simple linear regression, where previous calculation will influence on results of further calculations:

$$f(\text{node}[m]) = \sum_{n=1}^{m-1} (\text{clear}_{f_{\text{function}}(\text{node}[m-n])})$$

### 3. CHOOSING ALGORITHM

When we talk about algorithm, which will suit everyone user, we mean, that there will be some fuzzy logic based system, which will choose, which function will suit one or another case. For this purpose, it's better to take a look at classification systems. For our case, we will take naïve bayes:

$$\text{classify}(f_1, \dots, f_n) = \arg \max p(C=c) \prod_{i=1}^n p(F_i = f_i | C = c)$$

Where, main option, by which we will take decision, is time set. If we go further, we will notice, that avg concentration depends from time set. Every timezone, by which certain prediction rule changes – is called frame. So, the idea comes to defining these frames. Assume, that we already have collected data, and has certain rule, which helps us to define, what kind of frame do we have – short or long-term. For instance, let's assume, that for person A, the frame with length or 5 minutes – is a short-term, while frame with 20 minutes – Is long term. The key-determination option, by which we will choose one or another implementation of linear regression, is the curve of calculated concentration by time (one item per second). In simple words – while curve is smooth, we can treat this frame as a short one, otherwise – it should be a long frame. Let's generate an array of 1000 points, where each point – is a time stamp, and it's value is concentration level, received at this time (the code is written in Javascript ES6):

```
const svm = require('node-svm'),
    _ = require('lodash'),
    ml = require('shaman');

let stamp = new Date().getTime();

let xor = _.map(new Array(1000), (el, i) => [[i, stamp += 1000], _.random(10, 20)]);

let clf = new svm.SVM({
  svmType: 'NU_SVR',
  c: 1,
  kernelType: 'LINEAR',
  kFold: 4,
  normalize: true,
  reduce: true,
  cacheSize: 200,
  shrinking: true,
  probability: false
});

Promise.all([
  new Promise(res => {
    clf.train(xor).done(() => {
      res(clf.predictSync([1100, stamp + 1000]))
    })
  }),
  new Promise((res, rej) => {
    let x = _.map(xor, s => _.get(s, '0.1'));
    let y = _.map(xor, s => _.get(s, '1'));
  })
]);
```

```

let lr = new ml.LinearRegression(x, y);
lr.train((err)=> {
  err ? rej(err) : res(lr.predict(stamp + 1000));
});
})
)
.then(data=> {
  console.log('svm: ', _.head(data));
  console.log('linear: ', _.last(data));
});

```

Here, svm linear regression (NU SVR) and general linear regression won't differ much: svm's 14.999999386575372 vs linear's 14.855842543125618.

Now, let's make a decrease points in curve:

```

let xor = _.map(new Array(1000), (el, i)=>
  [[i, stamp += 1000], i > 300 && i < 600 ? _.random(10, 20) : _.random(80, 100)]
);

```

The result will be: svm - 56.998159564945304, linear - 73.75782167539. As you see, svm take into a count 'reduced points', while general linear regression oriented much on last received points.

#### 4. CONCLUSIONS

In this paper, we outlined a way, in which we can make a long-term and short-term predictions of mind state. We've provided a quick walkthrough of prediction's problems. Also, we've proposed a model, by which we can increase accuracy of calculated average concentration level thanks to using previous node's value in new calculations, and choosing the algorithm itself, by which certain frame will be calculated.

In next article we will devote ourselves in problems related to the nature of data, which we collect from EEG device. We will have a look at patterns and artefacts in terms of neurology. And dive into issues related to noise in data and anomalies.

Finally, I would like to admit, that subject, which we've covered in this article, is not so clear from the first point of view, as it requires more complex expert system, the proposed model could be a good start, which we are going to develop and write about it alongside our research.

#### *Список литературы / References*

1. *Фреунд Осуна Р. и Гирози Ф.* Продвинутый учитель для SVM. Д. Принцип, Л. Джиль, Н. Морган, и Е. Уилсон, авторы. Нейронные сети и обработка сигналов. VII — конференция, 1997 IEEE Workshop. С. 276 – 285. Нью Йорк, 1997. IEEE.
2. *Абу-Мостафа Б. С.*, 1990. Обучение по совпадениям в нейронных сетях, Д. Сомплекиты. С. 192–198.
3. *Тибишрани Р. & Фриедман Д.* (2008), Элементы статистического обучения. Спрингер. Калифорния.
4. *Акаике Х.*, 1970. Идентификация статистического предиктора. Институт Статической математики. 22 ежегодная конференция. С. 203–217.
5. *Ангел Д. Р. П., Уизинович П., Ллойд-Харт М. и Сандлер Д.*, 1990. Адаптивная оптика для массивов телескопов, с использованием нейронных сетей. Nature 348. С. 221–224.
6. *Баум Е. Б.* (1991). Нейронные сети, которые обучаются в полиномином времени, на основе примеров и запросов. IEEE Trans. On neural networks 21. С. 5–19.
7. *Уестреич Д., Лесслер Д. & Ёнссон М.*, 2010. Достижение оптимальных результатов: нейронные сети, SVM, decision trees (CART) и мета-классификаторы как альтернатива логической регрессии. Journal of Clinical Epidemiology 63. 826–833.
8. *Икбал М. С., Мисра Х. и Беганараыана Б.* Анализ само-ассоциативных нейронных сетей. IEEE Международная конференция по нейронным сетям. Вашингтон. США, 1999. С. 70.
9. *Щимке С., Виелхауер С., Диттманн Д.* Использование адаптивной дистанции Левенштейна для онлайн аутефикации. Распознавание паттернов, 17 международная конференция (ICPR'04), 2004.

10. Давидсон Р. Д. и Лутз А. «Разум Будды: Нейропластичность и медитация». IEEE Signal Processing 2007. С. 171 - 174.
11. Валле Роналд С. и Ёхн М. Левине. «Эффекты наблюдения за контролем альфа волн». Психофизиология 12.3, 1975. С. 306-309.
12. Лотте Ф., Сонгедо М., Лесуьер А., Ламарче, Ф. & Арналди Б., 2007. Июнь. Обзор алгоритмов классификации для ВСИ интерфейсов. Журнал нейронной инженерии. 4 (2). R1–R13.

*Список литературы на английском языке / References in English*

1. Osuna R. Freund and. Girosi F. An improved training algorithm for support vector machines. In J. Principe, L. Gile, N. Morgan, and E. Wilson, editors, Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop. Pages 276 – 285. New York, 1997. IEEE.
2. Abu-Mostafa Y. S. (1990). Learning from hints in neural networks, J. Complexity 6, 192–198.
3. Tibshirani, R. & Friedman, J. (2008), The Elements of Statistical Learning, Springer, California
4. Akaike H. (1970). Statistical predictor identification, Ann. Inst. Statist. Math. 22, 203–217.
5. Angel J. R. P., Wizinowich P., Lloyd-Hart M. and Sandler D., 1990. Adaptive optics for array telescopes using neural-network techniques. Nature 348, 221–224.
6. Baum E. B. (1991). Neural net algorithms that learn in polynomial time from examples and queries, IEEE Trans. on neural networks 2 1, 5–19.
7. Westreich D., Lessler J. & Jonsson M., 2010. Propensity score estimation: Neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. Journal of Clinical Epidemiology 63, 826–833.
8. Ikbal M. S., Misra H. and Yegnanarayana B. Analysis of autoassociative mapping neural networks. In IEEE Proceedings of the International Joint Conference on Neural Networks, Washington. USA, 1999. 70.
9. Schimke S., Vielhauer C., Dittmann J. Using Adapted Levenshtein Distance for On-Line Signature Authentication. Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04), 2004.
10. Davidson R. J. and Lutz A. “Buddha’s Brain: Neuroplasticity and Meditation”. IEEE Signal Processing, 2007: 171 - 174. Print.
11. Valle Ronald S. and Levine John M. “Expectation Effects in Alpha Wave Control”. Psychophysiology 12.3 (1975): 306-309. Print.
12. Lotte F., Congedo M., Lecuyer A., Lamarche F. & Arnaldi B., 2007. June. A review of classification algorithms for EEG-based brain-computer interfaces. Journal of neural engineering. 4 (2). R 1–R 13.