Radboud University Nijmegen

Bachelor Thesis

Artificial Intelligence

Classifying cognitive load from galvanic skin response time domain features

By:
Guido Zuidhof
guido.zuidhof@student.ru.nl

Supervisors:
Louis Vuurpijl
Pashiera Barkhuysen
Ervin Poljac

January 11, 2015

Contents

Abstract

To do.

Chapter 1

Introduction

To do: Short introductory story

1.1 Cognitive load

Cognitive load theory [Sweller, 1994] revolves around the idea that (short-term) working memory is limited, whereas long-term memory is unlimited.

It is built upon *schema theory*, in which expertise in an area is a function of the acquisition of particular schemata [Bartlett and Bartlett, 1995]. A schema is a mental structure to organise knowledge. [McVee et al., 2005] provides a review of schema theory.

When processing information, we use working memory. A schema is one element in this working memory, no matter how large it is [Mayer, 2014].

Cognitive load theory is nearly always studied in the context of learning new schemata. The idea is that there are ways to reduce cognitive load by utilising long-term memory, and thus having more space in working memory, which would allow for more efficient learning.

Cognitive load theory distinguishes three types of cognitive load, which according to research can each be reduced in specific ways [Mayer, 2002]:

Extraneous cognitive load This cognitive load is influenced by how the material or task is presented. It can be reduced by using worked-out examples, using diagrams, or presenting the material in multiple modalities [Mousavi et al., 1995].

Intrinsic cognitive load This cognitive load is determined by the complexity of the material at hand. It can be reduced by splitting up the task or building upon former (informal) knowledge.

Germane cognitive load Germane cognitive load is the cognitive load associated with the process of building, acquiring and automating new schemas.

The greater the space in working memory for this load (by reducing the intrinsic and extraneous cognitive load), the greater the ease of learning.

1.1.1 Applications

There are many applications where knowing the cognitive load of a person yields valuable information. Here I will highlight two:

Learning Most applications and research regarding cognitive load theory is applied in the context of multimedia learning [Brunken et al., 2003]. The goal in this domain is to find the way to structure and present information to ease learning the most. In other words, to reduce the intrinsic and extraneous cognitive load as much as possible, so that more working memory is left for germane cognitive load.

Usability testing Cognitive load measures, and more generally stress and arousal measures, are used by human computer interaction (HCI) developers and researchers to evaluate the usability of software or systems [Jacob and Karn, 2003] [Schmutz et al., 2009].

[Schmutz et al., 2009] investigated cognitive load measurement in web applications. Concluded was that cognitive load measures can be used as a *"valuable additional measure of efficiency"*, which strongly correlated to general satisfaction of the user.

1.2 Determining cognitive load

To do: How this cognitive load could be determined, i.e. by physiological measurement. [Brunken et al., 2003] is a good survey

1.3 Physiological response to cognitive load

To do: How a human body responds to cognitive load physiologically, why this measurement method/approach is a good idea.

1.3.1 Galvanic Skin Response

Galvanic Skin Response (GSR) is an interesting physiological response to measure as sensors are cheap and simple, and the subject is not or marginally restricted in movement. To this end GSR is the biometric of choice in this explorative study.

For this reason mobile GSR measurement systems, for instance around the foot [Gravenhorst et al., 2013] are being developed to measure (patients) throughout the day. Also, new devices such as smartwatches are fitted with

GSR electrodes increasingly often, which means findings could be applied there. An example would be using GSR data measured by a smart watch to dynamically change an application's presentation of information.

GSR can be measured in two ways, by measuring conductance (in siemens) or resistance (in ohms). Measuring skin conductance *"bears a simpler more linear relationship to the underlying processes of psychological interest than its repriprocal, SR [skin resistance]"* [Lykken and Venables, 1971], therefore any GSR measurement in this thesis is done by measuring skin conductance.

1.4 Research aim

The goal of this thesis is to investigate the possibility of determining what the cognitive load of a person is, given GSR measurements. An explorative experiment will be used with different tasks, each of which evokes a different level of cognitive load. From this data a classifier is built, which can classify which task the user was doing based on features derived from GSR, and thus say something about the cognitive load of a person.

1.5 State of the art

Galvanic skin response readings have been studied in regard to cognitive load.

[Shi et al., 2007] investigated the possibility of using GSR as an index of cognitive load during the use of unimodal and multimodal versions of the same interface. It has been shown that multimodal interfaces *"support the user in managing cognitive load"* [Oviatt et al., 2004], and thus multimodal interfaces have a lower corresponding cognitive load.

Shi et al. found that GSR values increase when cognitive load levels increase across all subjects, and conclude that GSR can be used to "serve as an objective indicator of [a] user's cognitive load".

[Nourbakhsh et al., 2012] showed that both time and frequency domain features from GSR data significantly correlate with the cognitive load of a person.

In the follow-up research GSR and blink features were used for cognitive load classification [Nourbakhsh et al., 2013]. Combining these two feature-sets, around 75% accuracy for binary classification, and 50% accuracy for four-class classification was achieved. Only GSR features lead to accuracies of around 35% and 66% for binary and 4-class classification resepctively.

Chapter 2

Method

2.1 Experiment

The experiment is mostly based on the arithmetic task experiment conducted by Nourbakhsh et al. [Nourbakhsh et al., 2012]. The experiment involves adding up four numbers, displayed one by one, and selecting the right from three possible answers.

There are four difficulty levels of this task. Difficulty one involves adding up four binary numbers (1 and 0). Difficulty level two, three and four consists of adding up four numbers with a length of one, two and three digits respectively.

The subject is first shown an amount of stars corresponding to the difficulty level for 8 seconds, after which the numbers are shown one by one for 4 seconds. Three answers are then presented on the screen, of which one is correct. The subject then clicks the correct answer using a standard computer mouse. There was no time limit for selecting an answer.

In [Nourbakhsh et al., 2013] these tasks were shown to have significant difference in subjective ratings of task difficulty levels, as well as a siginificant difference in response time of different task levels. From which is shown that these designed tasks manipulate cognitive load effectively.

2.1.1 Apparatus

The task was displayed in a 15.6” laptop screen, input was given using a standard computer mouse. Two galvanic skin response sensors were used in this experiment. The participants wore both at the same time on the left arm. Below, the two sensors are briefly discussed.

Affectiva Q Sensor This sensor is embedded in a wrist band, it uses a dry electrode. The sensor side of the wrist was placed on the bottom of the wrist. The sampling rate of this sensor was set to 8 samples per second.

BIOPAC MP30 The BIOPAC MP30 was used in conjunction with finger GSR sensors. This sensor wraps around two fingers, in this experiment the index and ring finger. It uses gel electrodes, Grass EC33 Electrode Paste was used for this purpose.

The sampling rate was set to 1000 samples per second.

2.1.2 Participants

Six participants took part in this experiment, five male and one female. Their ages ranged between 19 and 25 at the time of the experiment. They were not compensated for their participation.

2.1.3 Task

2.1.4 Experimental Design

The task consisted of 8 arithmetic tasks, with 4 difficulty levels. Three of every difficulty level were completed by the participants, in a random order.

The task consisted of adding up three numbers, and selecting the right answer from three options. The numbers to add were shown one by one, for four seconds. Before a task starts, a number of stars is shown equal to the difficulty level, for eight seconds. The tasks followed eachother without breaks. There was no time limit for selecting the answer, there was no feedback whether the selected answer was correct.

2.1.5 Procedure

The participants were first explained the task and seated in front of the laptop. Then the sensors were applied to the left hand and wrist. The participant was then instructed to place the left hand on the table in front of them and not to move it, in order to prevent possible motion artifacts which have been shown to influence GSR readings [**?**].

2.2 Analysis

2.2.1 Affectiva Q Sensor Data

The data gathered from the Affectiva Q Sensor proved unusable. In most participants no proper contact seemed to have been made with the skin during the full experiment. Dry GSR sensors need some sweat to ensure a good connection with both electrodes. Given the short experiment duration (a few of minutes) this connection was likely never made.

In hindsight the participants could have worn the device longer before the first task start, or they could have been asked to perform some light exercise.

Any data discussed from here on is solely that collected using the BIOPAC MP30 and it's finger electrodes.

2.2.2 Preprocessing

All data files were preprocessed using custom scripts written in *MATLAB (The MathWorks, Inc.)*. The collected data contained a DC current, which was smoothed by filtering and resampling to 100 samples per second using a Savitzky-Golay filter [**?**].

The data per subject was then cut into seperate frames, the tasks. A task (slice of data) runs from the time that the first number is shown, to the time that an answer is given. These tasks were then smoothed one last time, to get rid of a small remaining amount of DC noise. This was again done using a Savitzky-Golay filter.

The data was then normalized per subject over all datapoints in all tasks. This was done by dividing every data point of the subject by the average of the subject over all tasks.

Let the average over all tasks:

$$AverageGSR(s) = \frac{\sum_{q \in Q_s} \sum_{t \in T_q} GSR(s,q,t)}{\sum_{q \in Q_s} |T_q|}$$

With s a given subject, Q_s a set of all tasks performed by s, and T_q the set of all measurements in task q, and $GSR(s,q,t)$ be the GSR datapoint of subject s, task q at time t.

The normalised GSR is then given by:

$$NormGSR(s,q,t) = \frac{GSR(s,q,t)}{AverageGSR(s)}$$

Note that the normalization is done by dividing by the average of the subject over all tasks, and not by simply transforming all measurements to be between 0 and 1. What (original) GSR value would become 1 and 0 would then be quite arbitrary per subject, and not all subjects would be on the same scale, as outliers can totally influence this range.

This normalised GSR was then cut into the specific tasks.

2.2.3 Transformation

From the measurements from every task certain time domain features were extracted.

The extracted features:

- Average over all points in the task.
- Accumulative of all points in the task.

- Standard deviation of all points in the task.
- Difference between first and last point of the task.
- Peaks with varying tresholds of what a peak is.

Below I will elaborate how these features were calculated from the data. In these formula's s is a subject, q is a task and T_q are all the measurements (datapoints) associated with task q.

Average

$$AvgTaskGSR(s,q) = \frac{\sum_{t \in T_q} NormGSR(s,q,t)}{|T_q|}$$

Accumulative

$$AccTaskGSR(s,q) = \sum_{t \in T_q} NormGSR(s,q,t)$$

Standard deviation

$$StdDevTaskGSR(s,q) = \sqrt{\frac{1}{|T_q|} \sum_{t \in T_q} (NormGSR(s,q,t) - AvgTaskGSR(s,q))^2}$$

Difference

$$DiffTaskGSR(s,q) = NormGSR(s,q,t_{|T_q|}) - NormGSR(s,q,t_1)$$

where t_1 is the first measurement in T_q, and $t_{|T_q|}$ the last measurement of the task.

Peaks For peak detection the *PeakFinder* script [Yoder, 2009], found in the *Matlab Central File Exchange* was used under BSD License. This script finds local maxima or minima, even in noisy signals, and can be supplied with a treshold that determines what classifies as a peak.

This script was called with four different threshold levels, given by

$$Threshold(s,q,k) = \frac{\max_{t \in T_q} NormGSR(s,q,t) - \min_{t \in T_q} NormGSR(s,q,t)}{(2k)^2}$$

with $k \in \{1,2,3,4\}$.

2.2.4 Data mining

These extracted features are then fed into *Weka 3* [Hall et al., 2009], which is data mining software in Java.

Three of the most commonly used classification algorithms were used [Ruparel et al., 2013]. Namely a Bayesian classification algorithm (*NaïveBayes*, which is an implementation based on [John and Langley, 1995]), a support vector machine (SVM) classification algorithm (*LibSVM*) [Chang and Lin, 2011] and a neural network classifier (*MultilayerPerceptron*, with 8 hidden layers).

Two classifiers were trained with each algorithm, one for 4-class classification, and one for 2-class (binary) classification (with task difficulty 1 and 2 combined, and 3 and 4 combined).

Chapter 3

Results

3.1 Classification accuracy

Table 3.1: Classication accuracy

Classification Algorithm	2-Class Classification	4-Class Classification
NaiveBayes	62.5%	35.4%
LibSVM	68.8%	22.9%
MultilayerPerceptron	56.3%	41.7%

The performance of each predictive model was estimated using 10 fold cross-validation. The percentages in the table are the percentage of correctly classified task difficulties by the predictive model.

3.2 Confusion matrices

Table 3.2: NaiveBayes confusion matrix

1	2	3	4	<- Classified as difficulty
1	7	1	3	Difficulty 1
3	7	0	2	Difficulty 2
2	4	2	4	Difficulty 3
2	2	1	7	Difficulty 4

Table 3.3: LibSVM confusion matrix

1	2	3	4	<- Classified as difficulty
1	5	6	0	Difficulty 1
5	3	3	1	Difficulty 2
3	1	5	3	Difficulty 3
5	0	5	2	Difficulty 4

Table 3.4: MultilayerPerceptron confusion matrix

1	2	3	4	<- Classified as difficulty
3	4	3	2	Difficulty 1
3	5	2	2	Difficulty 2
2	3	6	1	Difficulty 3
2	1	3	6	Difficulty 4

Chapter 4

Discussion

4.1 Interpretation

The results show that the classification accuracy of all three classifiers are better than chance dictates, however not much. In the binary classification task the results were more reasonable. These classification rates are slightly lower than those reported by [Nourbakhsh et al., 2013] in a similar experiment, this may be due to the smaller dataset.

When considering the confusion matrices of the 4-class classification it can be noted that most wrongly classified task difficulties was task 1, which was more often classified as difficulty 2 and 3 than 1. In fact, it has only been classified correct once out of twelve entries in the NaiveBayes and LibSVM model, and a mere three times in the MultilayerPerceptron model. Perhaps this task was too easy and caused participants to become distracted.

There are many possible explanations why this classification accuracy is not greater than it is. The first explanation is that from just GSR data, it simply can not be predicted any better. Next I will highlight some other possible reasons:

Loss of information from filter

Filtering the data, to remove the DC current noise, lead to some slight loss of data. It smoothed peaks slightly, which may have influenced the classification rate.

Task bleeding

Another explanation may be the bleeding of tasks into other tasks. If for instance a subject first gets two hard tasks, and then an easy one, it is imaginable that the subjects GSR value for that easy task is higher than it would have been without first having done harder tasks.

This is mitigated by the random order in which the tasks are presented and the time between tasks, however with the relatively small subject pool it is not unimaginable this has had its effect on the classification accuracy.

Dataset size

With 6 subjects and 8 tasks per subject, the total dataset consisted of 48 entries. Training a classification algorithm on a small dataset may cause outliers to influence it to fairly great effect.

Task cognitive load manipulation

Despite [Nourbakhsh et al., 2013] showing that the tasks effectively manipulate the cognitive load of the subject, this was concluded from subjective reports and response times. It may be that these response times were manipulated by something other than cognitive load, and that the tasks do not evoke a difference in cognitive load in the subject. It may also be that this difference in cognitive load is very small, making classification harder, especially with more than two classes.

Frequency domain features

All extracted features were features from the time domain, frequency domain features were shown by [Nourbakhsh et al., 2012] to significantly correlate with the task difficulty as well. Adding frequency domain features to the featureset, may improve classification rates.

4.2 Future research

Future research could be measuring GSR data using longer, carefully designed tasks, of which the cognitive load can be determined using an established cognitive load index. Also future research could include investigate how classification accuracy changes when using data measured by more portable (consumer) sensors, whose measurements are likely of lesser quality than the used lab equipment in this experiment.

Bibliography

F. C. Bartlett and F. C. Bartlett. *Remembering: A study in experimental and social psychology*, volume 14. Cambridge University Press, 1995.

R. Brunken, J. L. Plass, and D. Leutner. Direct measurement of cognitive load in multimedia learning. *Educational Psychologist*, 38(1):53–61, 2003.

C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011. ISSN 2157-6904. doi: 10.1145/1961189.1961199. URL http://doi.acm.org/10.1145/1961189.1961199.

F. Gravenhorst, A. Muaremi, G. Tröster, B. Arnrich, and A. Gruenerbl. Towards a mobile galvanic skin response measurement system for mentally disordered patients. In *Proceedings of the 8th International Conference on Body Area Networks*, BodyNets '13, pages 432–435, ICST, Brussels, Belgium, Belgium, 2013. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). ISBN 978-1-936968-89-3. doi: 10.4108/icst.bodynets.2013.253684. URL http://dx.doi.org/10.4108/icst.bodynets.2013.253684.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278. URL http://doi.acm.org/10.1145/1656274.1656278.

R. J. Jacob and K. S. Karn. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4, 2003.

G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

D. T. Lykken and P. H. Venables. Direct measurement of skin conductance: A proposal for standardization. *Psychophysiology*, 8(5):656–672, 1971.

R. Mayer. *The Cambridge Handbook of Multimedia Learning.* Cambridge Handbooks in Psychology. Cambridge University Press, 2014. ISBN 9781139992480. URL `https://books.google.nl/books?id=Cvw6BAAAQBAJ`.

R. E. Mayer. Multimedia learning. *Psychology of Learning and Motivation*, 41:85–139, 2002.

M. B. McVee, K. Dunsmore, and J. R. Gavelek. Schema theory revisited. *Review of educational research*, 75(4):531–566, 2005.

S. Y. Mousavi, R. Low, and J. Sweller. Reducing cognitive load by mixing auditory and visual presentation modes. *Journal of educational psychology*, 87(2):319, 1995.

N. Nourbakhsh, Y. Wang, F. Chen, and R. A. Calvo. Using galvanic skin response for cognitive load measurement in arithmetic and reading tasks. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pages 420–423, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1438-1. doi: 10.1145/2414536.2414602. URL `http://doi.acm.org/10.1145/2414536.2414602`.

N. Nourbakhsh, Y. Wang, and F. Chen. Gsr and blink features for cognitive load classification. In *INTERACT 2013*, page 8, Cape Town, South Africa, September 2013.

S. Oviatt, R. Coulston, and R. Lunsford. When do we interact multimodally?: Cognitive load and multimodal communication patterns. In *Proceedings of the 6th International Conference on Multimodal Interfaces*, ICMI '04, pages 129–136, New York, NY, USA, 2004. ACM. ISBN 1-58113-995-0. doi: 10.1145/1027933.1027957. URL `http://doi.acm.org/10.1145/1027933.1027957`.

N. H. Ruparel, N. M. Shahane, and D. P. Bhamare. Article: Learning from small data set to build classification model: A survey. *IJCA Proceedings on International Conference on Recent Trends in Engineering and Technology 2013*, ICRTET(4):23–26, May 2013. Full text available.

P. Schmutz, S. Heinz, Y. Métrailler, and K. Opwis. Cognitive load in ecommerce applications: Measurement and effects on user satisfaction. *Adv. in Hum.-Comp. Int.*, 2009:3:1–3:9, Jan. 2009. ISSN 1687-5893. doi: 10.1155/2009/121494. URL `http://dx.doi.org/10.1155/2009/121494`.

Y. Shi, N. Ruiz, R. Taib, E. Choi, and F. Chen. Galvanic skin response (gsr) as an index of cognitive load. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pages 2651–2656, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-642-4. doi: 10.1145/1240866.1241057. URL `http://doi.acm.org/10.1145/1240866.1241057`.

J. Sweller. Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4):295–312, 1994.

N. Yoder. Peakfinder, 2009. URL `http://www.mathworks.com/matlabcentral/fileexchange/25500-peakfinder`.