# Assessing the Degree of Nativeness and Parkinson's Condition Using Gaussian Processes and Deep Rectifier Neural Networks

*Tamás Grósz*[1], *Róbert Busa-Fekete*[2], *Gábor Gosztolya*[3], *László Tóth*[3]

[1]Institute of Informatics, University of Szeged, Hungary
[2]Department of Computer Science, University of Paderborn, Germany
[3]MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary

groszt@inf.u-szeged.hu, busarobi@upb.de, {ggabor,tothl}@inf.u-szeged.hu

## Abstract

The Interspeech 2015 Computational Paralinguistics Challenge includes two regression learning tasks, namely the Parkinson's Condition Sub-Challenge and the Degree of Nativeness Sub-Challenge. We evaluated two state-of-the-art machine learning methods on the tasks, namely Deep Neural Networks (DNN) and Gaussian Processes Regression (GPR). We also experiented with various classifier combination and feature selection methods. For the Degree of Nativeness sub-challenge we obtained a far better Spearman correlation value than the one presented in the baseline paper. As regards the Parkinson's Condition Sub-Challenge, we showed that both DNN and GPR are competitive with the baseline SVM, and that the results can be improved further by combining the classifiers. However, we obtained by far the best results when we applied a speaker clustering method to identify the files that belong to the same speaker.

**Index Terms**: Computational Paralinguistics, Challenge, Parkinson's Condition, Degree of Nativeness, Deep Neural Networks, Gaussian Processes

## 1. Introduction

The Interspeech 2015 Computational Paralinguistics Challenge (ComParE) deals with states of speakers as manifested in their speech signal's acoustic properties. Most of the tasks belonging to this paralinguistic area are classification ones; however, there are regression tasks as well such as estimating the age [1] or alcohol intoxication level [2, 3] of the speaker, or the intensity of conflict present [4, 5].

This year's Challenge [6] includes two regression tasks, which can be approached in a similar way. In the first one (the Parkinson's Condition (**PC**) Sub-Challenge), the neurological state of Parkinson patients had to be estimated according to the Unified Parkinson's Disease Rating Scale (UPDRS [7]), using utterances recorded at the Universidad de Antioquia in Colombia [8]. In the Degree of Nativeness (**DN**) Sub-Challenge, the pronunciation quality of non-native utterances has to be assessed, based on prosodic annotations [9]. This is a cross-corpus regression learning task, meaning that the training, development and test sets have different recording conditions (microphone, noise level, etc.); furthermore, their annotations are also on different scales. To this end, model prediction evaluation for both tasks is not performed via the common Pearson's correlation, but the organizers chose Spearman's correlation instead, which considers only the order of predictions.

Nowadays Gaussian Processes Regression is regarded as one of the state-of-the-art methods for regression in the machine learning community [10, 11, 12]. The speech technology community, however, prefers Artificial Neural Networks (ANN), especially since the invention of Deep Neural Networks (DNN) [13, 14]. In this study, submitted for ComParE 2015, we apply both methods. To improve their performance, we also try to combine them, and for the PC sub-challenge we also seek to identify the speakers (patients) to estimate the UPDRS scores of their utterances jointly.

## 2. Deep Rectifier Neural Networks

The core concept of deep networks is simple: build neural networks with many hidden layers instead of just one. Unfortunately, these deep neural networks are hard to train with standard SGD methods, so several algorithms have been proposed for their training. The first attempts focused on various pretraining strategies (e.g. [15, 16]), while it was shown recently that rectifier DNNs can attain a comparable performance even when trained with the standard backpropagation algorithm [17]. In deep rectifier neural networks, rectified linear units are employed as hidden neurons, which apply the rectifier activation function $max(0; x)$ instead of the usual sigmoid one [18].

In our previous studies we found that deep rectifier networks achieve better results on various tasks than other deep learning methods [19, 20], which was also confirmed for the current tasks by our preliminary tests. To train our DNN on a regression task, we applied linear output units with the MSE error function, and we employed L1 weight normalization for regularization. During our experiments we found that different network structures work best for the different tasks. For the PC task we trained DNNs with five hidden layers and 1000 neurons in each hidden layer, while for the DN task we used three hidden layers with 2000 hidden neurons in each.

To get an optimal performance from a neural network on a specific learning task, we need to fine-tune the metaparameters (network structure, learning rates, etc.) of the algorithm. To do this, first we trained 20 networks using just the training set, and evaluated their performance on the development set. Having found the best parameters, 50 DNNs were trained on the joint training and development sets, and the resulting nets were evaluated on the test vectors.

## 3. Gaussian Processes Regression

Gaussian Processes regression (GPR) [21, 22] is a nonparametric regression method. It means that it requires no parametric assumption about the form of the function to be learned, only some prior knowledge about its range and smoothness. It is not alone in the family of non-parametric methods: it can actually be related to kernel regression as well as to spline fitting.

The main ingredient (*input* or *prior*) of GP regression is the *kernel* function $K(x, x')$. It is a function of two observables (feature vectors) $x$ and $x'$, and it expresses how much the two corresponding targets $y$ and $y'$ are correlated:

$$\mathrm{Cov}\left\{y, y'\right\} = K(x, x'). \tag{1}$$

$K$ is symmetric and goes to 0 as the distance between $x$ and $x'$ grows, representing the intuition that targets become uncorrelated as the observables move far apart. $K$ is usually (but not necessarily) a monotonically decreasing function of the distance $d = |x' - x|$, expressing shift-invariance (stationarity) and the intuition that $y$ and $y'$ are more correlated if $x$ and $x'$ are close.

In this paper, we shall use the squared exponential (Gaussian) kernel

$$K(d) = a^2 \exp\left(-\frac{d^2}{w^2}\right). \tag{2}$$

Formally, this kernel implies that the fitted function be very smooth (infinitely differentiable). The *amplitude* parameter $a$ determines the range of the process (i.e. the range of target values), while the *width* parameter $w$ is the most important hyperparameter: it determines the smoothness of the process. The larger the value of $w$, the smoother the fit.

For a fixed kernel, a Gaussian Process (GP) represents a probability distribution over an infinite number of functions. Informally, one can think about a function $f(x)$ as a limit of a series of vectors indexed by $\big(f(x_1), \dots, f(x_n)\big)$ with $n \to \infty$. A GP is the limit of a multidimensional Gaussian over $\big(f(x_1), \dots, f(x_n)\big)$ when the dimension $n$ goes to infinity. Formally, it is a collection of random variables, any finite number of which have a joint Gaussian distribution [21]. Given a mean function $m(x)$ and a kernel $K(d)$, the GP is completely defined by

$$\mathbb{E}\left\{f(x)\right\} = m(x) \tag{3}$$

and

$$\begin{aligned}\mathrm{Cov}\left\{f(x), f(x')\right\} &= \mathbb{E}\left\{\big(f(x) - m(x)\big)\big(f(x') - m(x')\big)\right\} \\ &= K(|x - x'|).\end{aligned}$$

Throughout this paper, we set the prior mean function $m(x)$ to zero. This choice expresses the fact that outside of the observation range the fitted function is 0. In practice, a usual preprocessesing step is to center the target variable by subtracting its mean, which is equivalent to setting $m(x)$ to $\frac{1}{n}\sum_{i=1}^{n} y_i$.

In order to evaluate the GPR for a feature vector $x$, we condition the process on the observations by forcing it to be a Gaussian with parameters $\mathcal{N}(y_i, \sigma_i)$ at each $x_i$. What is nice about a GP is that it will remain a GP after conditioning it on the observations $D = \{(x_i, y_i)\}_{i=1}^{n}$. What will change is the mean function $\mathbb{E}\left\{f(x) \mid D\right\}$ and the variance function $\mathrm{Var}\left\{f(x) \mid D\right\}$. A second nice property of a GP is that both the mean function and the variance function can be computed analytically. Similarly to computing the conditional mean and variance of a multivariate normal distribution when some of the coordinates are fixed, we get

$$\begin{aligned}\mathbb{E}\left\{f(x) \mid D\right\} &= \mathbf{k}^T(\mathbf{K} + \Sigma)^{-1}\mathbf{y} &(4)\\ \mathrm{Var}\left\{f(x) \mid D\right\} &= K(0) - \mathbf{k}^T(\mathbf{K} + \Sigma)^{-1}\mathbf{k}, &(5)\end{aligned}$$

where $\mathbf{K} = \big[K_{i,j}\big]_{i,j=1}^{n} = \big[K(|x_i - x_j|)\big]_{i,j=1}^{n}$ is the kernel matrix, $\Sigma$ is a diagonal matrix with the squared error terms $\sigma_i^2$ in the diagonal, $\mathbf{y} = (y_1, \dots, y_n)$ is the vector of the targets, and $\mathbf{k} = \big(K(|x - x_1|), \dots, K(|x - x_n|)\big)$ is the vector of covariances between the inputs $x_i$ and the point of interest $x$ (where we are evaluating the mean and the variance functions).

We assumed constant noise for each instance, keeping $\sigma_i^2$ constant everywhere. In this way, there were three hyperparameters of the GPR: the amplitude of the kernel $a$, the width of the kernel $w$ and the noise parameter $\sigma$. In principle, the three GPR hyperparameters can be estimated by maximizing the marginal likelihood on a hold-out dataset [21]. But as our goal was to obtain a diverse pool of many regressors, we opted for running the GP regression with randomly chosen hyperparameters with 100 repetitions. We generated the logarithm of the parameters uniformly at random from the interval $[-1, 3]$. As a final step, we filtered out the models with very poor performance.

# 4. Feature Selection

We performed our experiments on the 6373-sized feature set extracted by the Challenge organizers, which is naturally full of redundant and irrelevant features. Although current state-of-the-art machine learning methods are able to make reliable predictions in this extremely high-dimensional space, it was shown that they can be assisted by feature selection in paralinguistic tasks as well [23, 24]. Therefore we decided to also carry out some kind of feature selection. However, as this study focuses on the application of DNNs and GPR for these regression tasks, and we did not want to waste our number of trials on the test set with parameter tuning for another sub-procedure, we opted for a quite simple method, hoping that it would be robust enough.

Our feature selection approach was based on the assumption that features which correlate well with our target score could be of more help for any machine learning algorithm. To this end, we calculated the Pearson's correlation coefficient with the target score for all the 6373 attributes, and sorted the attributes according to the absolute value of this coefficient. Then we performed simple nu-SVR regression (using the LibSVM library [25]) utilizing the first $n$ most correlated features.

For the Parkinson's Condition sub-challenge we found that by using the most correlated 1000 features, we could improve the Spearman's correlation value on the dev set from .453 to .574. In the DN task, however, we found the optimal number of features to be 25. This is an extremely low value, and it is not surprising that neither DNNs nor GPR performed well on this very pure feature subset, therefore we decided not to use feature selection for this sub-challenge.

# 5. Aggregating the Predictors

We applied several methods with several hyperparameters, which resulted in a diverse pool of models. Instead of selecting the best model (normally done in a validation step), we opted for combining them into an ensemble predictor. Formally, assume a set of models denoted by $\{f_1, \dots, f_m\}$, and each model $f_j$ computes a score value $s_i$ for each instance $x_i$ where $1 \le i \le n$. The simplest way to combine the scores of the models is to average them for each instance $x_i$. This approach is referred to as Average Scoring (AS).

Since the preferred evaluation metric for both sub-challenges was Spearman's correlation, which considers only the order of the output values, we also applied a rank-based aggregation of the output scores, namely the Borda method (BM) [26]. Formally, $r_{i,j}$ denotes the number of instances whose score is smaller than $x_i$ with respect to $f_j$, i.e. $r_{i,j} = \#\{1 \le \ell \le n | f_j(x_i) > f_j(x_\ell)\}$. Then the Borda score of $x_i$

| Method | | Dev | Test |
|---|---|---|---|
| DNN AS | **Spearman** | .425 | **.510** |
| | Pearson | .435 | .516 |
| DNN BM | **Spearman** | .429 | .506 |
| | Pearson | .430 | .509 |
| Baseline | **Spearman** | .415 | .359 |
| | Pearson | .403 | – |

Table 1: The results achieved with the different aggregating methods for the DN Sub-Challenge.

with respect to $\{f_1, \ldots, f_m\}$ is computed using

$$s_i^B = \frac{1}{m} \sum_{j=1}^{m} r_{i,j} \ .$$

The Spearman and Pearson correlations are computed with these combined AS and BM scores.

## 6. Results

As is standard in machine learning, first we fine-tuned the parameters of our methods on the development data and evaluated them on the test set only after having found the best parameters. The baseline study shows that SVM could perform better on the test set with a complexity parameter that is different from the one found optimal on the development set [6]. We will disregard this information here, because we consider this to be the result of an unfair peeking.

### 6.1. Degree of Nativeness Challenge

The speech material [9] comprises 5483 files, both the development and test set being disjunct from the training set and each other with respect to both speakers and sentences. As the data was collected from multiple databases, we needed to standardize the datasets independently to be able to train on both of them. Furthermore, even the regression labels were at different scales for the training and development data, so we needed to unify them; we re-scaled them to $[0, 1]$. This way, we were able to use both the training and development data to train the final models.

GPR could not learn meaningful models on this task: it achieved a correlation of .170 on the development set, which is far below the baseline; therefore we did not consider using it for this task. The reason might be that GPR overfit the training set, and could not learn a sufficiently general model from the data. It is not that surprising, though, considering that the development and the training sets were taken from completely different databases.

Table 1 shows the correlation results achieved by the DNN using the different voting methods. The DNN performed slightly better than the baseline SVM on the development set, but achieved much better results on the test set. On the test set, the DNN achieved a Spearman correlation value of .510, which is a 42% relative improvement compared to the baseline, and also significally exceed the best score (.425) reported in [6]. This improvement suggests that DNNs were able to learn more general models from the merged training and development sets.

As for the voting methods, the Borda voting technique offered some improvement on the development set, yet on the test set it performed slightly worse than the averaging method. The differences, however, are not significant in either case.

| Method | | Dev | | Test | |
|---|---|---|---|---|---|
| | | Pe. | **Sp.** | Pe. | **Sp.** |
| DNN | AS | .574 | .560 | .163 | .306 |
| | BM | .577 | .559 | — | — |
| GPR | AS | .499 | .497 | .237 | .213 |
| | BM | .492 | .496 | — | — |
| DNN + GPR | AS | .580 | .564 | .187 | **.310** |
| | BM | .555 | .548 | — | — |
| DNN + FS | AS | .570 | .579 | — | — |
| | BM | .579 | .579 | .334 | **.311** |
| SVM (baseline) | | .346 | .492 | — | .236 |

Table 2: The results achieved by the methods applied on the PC Sub-Challenge.

| Method | | Dev | | Test | |
|---|---|---|---|---|---|
| | | Pe. | **Sp.** | Pe. | **Sp.** |
| DNN | AS | .671 | .671 | .603 | **.649** |
| | BM | .672 | .670 | — | — |
| GPR | AS | .666 | .661 | — | — |
| | BM | .677 | .598 | — | — |
| DNN + GPR | AS | .679 | .671 | — | — |
| | BM | .691 | .665 | — | — |
| SVM (baseline) | | .346 | .492 | — | .236 |

Table 3: The results achieved by the various methods on the PC Sub-Challenge after feature selection and speaker clustering.

### 6.2. Parkinson's Condition Challenge

The recordings of this Sub-Challenge were taken from the same dataset [8]. Yet the recording conditions of the test set differed from the training data significantly, as it was recorded in a different (noisier) environment. As the 42 recordings of a patient correspond to various tasks from uttering sustained vowels to whole monologues describing the subject's typical day [8], the length of the utterances also varied greatly (between 0.2 and 153 seconds), making all the sets very diverse.

As Table 2 shows, the DNN attained quite good results on the development set, yet on the test set there is a big performance drop. The reason for this might be the different noise conditions in the training and test sets, which caused our nets to overfit the clean train and development data, just like the baseline SVM did. Of course, due to the sound quality difference between the test and development sets, it was very difficult to train a method that could perform well on both sets.

GPR alone could not achieve a good Spearman correlation, but its Pearson score was high compared to that of the DNN case. Combining the output of GPR and DNNs improved the results both on the development and on the test sets. Applying feature selection further improved the scores a bit, especially the Pearson's correlation on the test set.

#### *6.2.1. Speaker Clustering*

The left hand side of Figure 1 compares the estimated scores of our best system (DNN + feature selection + BM) with those of the expert annotations on the development set. Even at first glance, it differs from the output of a standard regression task in that only some values appear in the annotation, but they apply for several examples (see the vertical lines). This is the specialty

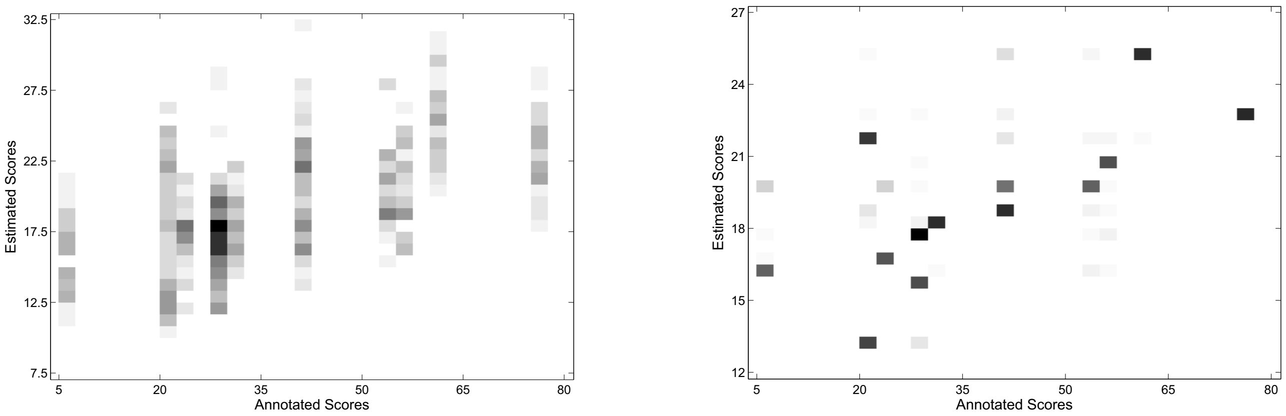

Figure 1: Density scatter plot of the annotated and estimated UPDRS scores without (left) and with (right) joint UPDRS estimation.

of this given dataset: the target scores were manually assigned scores of the patient, following the UPDRS-III standard. This procedure summarizes the state of *each patient* in *one* numerical value based on his speech and on a number of other motor functions such as facial expression and hand movement [7]; in the sub-challenge each of the 42 utterances of a patient had the UPDRS score of the patient [8]. It is clear that estimating the UPDRS score based on only one, sometimes very short recording (e.g. a sustained vowel) is an extremely difficult task.

However, in Fig. 1 we can also see the correlating trend of the real and the estimated scores (with the exception of a few speakers). If we could identify the utterances belonging to each speaker, we could estimate the score of these files jointly, hopefully leading to a better score estimation. Finding the utterances that belong to the same speaker is known as speaker clustering [27], which was shown to be useful in a number of computational paralinguistic tasks (e.g. [28, 29]).

Although in the PC task the speakers of the recordings were not revealed, we could easily identify them in the training and development sets by using the public information that all the 42 utterances of a patient had the same score. In a few cases multiple subjects shared the same score; we distinguished these speakers by the $F_0$ score of the recordings, hoping that this small imprecision would not hinder the clustering process.

We followed the approach of speaker clustering by feature selection: as various kinds of valid clusters can be formed in such a high-dimensional dataset, we turned to selecting those attributes which correlate well with speaker change. As the number of separate speakers was public in the training, development and test sets, the number of clusters was known beforehand. We utilized the K-means algorithm [30, 31], and relied on the entropy metric for clustering quality [32, 33]. We started with an empty set of selected features, and commenced an iterative process. For each iteration we expanded our set of chosen features with the next attribute; if we could achieve a better clustering on the training set, we kept the given feature, otherwise we discarded it. Next, we clustered the development and test sets by K-means, using only the features retained by this selection process. Finally, for each cluster we averaged out the UPDRS estimates of the appropriate utterances, and these averaged scores were used as the final estimates for each utterance in the cluster.

The right hand side of Figure 1 shows our estimates after the speaker clustering and averaging steps; the correlating trends of the UPDRS scores is much more convincing that it was on the left hand side. Table 3. shows the correlations got after feature selection and per-speaker averaging of the evaluated outputs. Both the DNN and GPR outperformed the baseline result, yet their combination did not offer any further improvement on the dev set, so we evaluated only the DNN on the test set. From our results, we achieved by far the best one with this configuration (.649) on the test set, significantly outperforming the value of .390 reported in the baseline paper. The corresponding Pearson's correlation value was also quite high.

The reason why speaker clustering improved the performance of our regression models so much might be that correlation, in contrast with the absolute difference of the ground-truth scores and their estimates, is only concerned with the tendency of the scores. It is especially true for the Spearman's correlation, which considers only the order of the estimates. As in the PC task there are a lot of equal scores in the annotation, correlation scores can be improved considerably if we force our algorithms to assign the same estimate to the elements of these groups as well. Our results indicate that this can be achieved efficiently by averaging the within-cluster estimates after speaker clustering. One of the difficulties of this task was that machine learning methods rarely optimize for any kind of correlation, but usually minimize some convex losses that are surrogate losses of some standard instance-based performance metric such as error rate.

## 7. Conclusions

We applied two state-of-the-art machine learning methods in the regression Sub-Challenges of the Interspeech 2015 Computational Paralinguistics Challenge: Deep Rectifier Neural Networks and Gaussian Processes Regression. Our results show that the DNN consistently managed to outperform the baseline SVM scores, while the performance of GPR varied to a significant extent. We experimented with two different output aggregation methods, and both of them produced quite good results. On the Parkinson's Condition Sub-Challenge we achieved the best results by using feature selection and by averaging out the scores of multiple recordings clustered to the same person.

## 8. Acknowledgements

This publication is supported by the European Union and co-funded by the European Social Fund. Project title: The denouement of talent in favour of the excellence of the University of Szeged. Project number: TÁMOP-4.2.2.B-15/1/KONV-2015-0006.

# 9. References

[1] M. Bahari and H. Van Hamme, "Speaker age estimation using Hidden Markov Model weight supervectors," in *Proceedings of ISSPA*, Montreal, Quebec, Canada, July 2012, pp. 517–521.

[2] B. Schuller, S. Steidl, A. Batliner, F. Schiel, and J. Krajewski, "The INTERSPEECH 2011 speaker state challenge," in *Proceedings of Interspeech*, Florence, Italy, Sep 2011.

[3] D. Bone, M. P. Black, M. Li, A. Metallinou, S. Lee, and S. S. Narayanan, "Intoxicated speech detection by fusion of speaker normalized hierarchical features and GMM supervectors," in *Proceedings of Interspeech*, Florence, Italy, Sep 2011, pp. 3217–3220.

[4] S. Kim, F. Valente, M. Filippone, and A. Vinciarelli, "Predicting continuous conflict perception with Bayesian Gaussian Processes," *IEEE Transactions on Affective Computing*, vol. 5, no. 2, pp. 187–200, 2014.

[5] B. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The Interspeech 2013 Computational Paralinguistics Challenge: Social signals, Conflict, Emotion, Autism," in *Proceedings of Interspeech*, Lyon, France, 2013.

[6] B. Schuller, S. Steidl, A. Batliner, S. Hantke, F. Hönig, J. R. Orozco-Arroyave, E. Nöth, Y. Zhang, and F. Weninger, "The interspeech 2015 computational paralinguistics challenge: Nativeness, Parkinson's & eating condition," in *Proceedings of Interspeech*, 2015.

[7] G. Stebbing and C. Goetz, "Factor structure of the unified Parkinson's disease rating scale: Motor examination section," *Movement Disorders*, vol. 13, pp. 633–636, 1988.

[8] J. R. Orozco-Arroyave, J. Arias-Londono, J. Vargas-Bonilla, M. González-Rátiva, and E. Nöth, "New Spanish speech corpus database for the analysis of people suffering from Parkinson's disease," in *9th Language Resources and Evaluation Conference (LREC)*, 2014, pp. 342–347.

[9] F. Hönig, A. Batliner, and E. Nöth, "Automatic assessment of non-native prosody - annotation, modelling and evaluation," in *International Symposium on Automatic Detection of Errors in Pronunciation Training (IS ADEPT)*, 2012, pp. 21–30.

[10] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process bandits: An experimental design approach," in *Proceedings of NIPS Workshop*, Whistler, BC. Canada, Oct 2009.

[11] R. Bardenet and B. Kégl, "Surrogating the surrogate: accelerating Gaussian-process-based global optimization with a mixture cross-entropy algorithm," in *Proceedings of ICML*, Haifa, Israel, June 2010, pp. 55–62.

[12] V. Lázár, I. Nagy, R. Spohn, B. Csörgő, A. Györkei, A. Nyerges, B. Horváth, A. Vörös, R. Busa-Fekete, M. Hrtyan, B. Bogos, O. Méhi, G. Fekete, B. Szappanos, B. Kégl, B. Papp, and C. Pál, "Genome-wide analysis captures the determinants of the antibiotic cross-resistance interaction network," *Nature Communications*, vol. 5, no. 4352, 2014.

[13] D. Yu and L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, 2014.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[15] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[16] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU*, 2011, pp. 24–29.

[17] L. Tóth, "Phone recognition with deep sparse rectifier neural networks," in *Proceedings of ICASSP*, Vancouver, Canada, 2013, pp. 6985–6989.

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of AISTATS*, 2011, pp. 315–323.

[19] T. Grósz and L. Tóth, "A comparison of deep neural network training methods for large vocabulary speech recognition," in *Proceedings of TSD*, Plzen, Czech Republic, 2013, pp. 36–43.

[20] G. Gosztolya, T. Grósz, R. Busa-Fekete, and L. Tóth, "Detecting the intensity of cognitive and physical load using AdaBoost and Deep Rectifier Neural Networks," in *Proceedings of Interspeech*, Singapore, Sep 2014, pp. 452–456.

[21] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[22] M. Seeger, "Gaussian Processes for Machine Learning," *International Journal of Neural Systems*, vol. 14, pp. 69–106, 2004.

[23] H. Kaya, T. Özkaptan, A. A. Salah, and F. Gürgen, "Canonical Correlation Analysis and Local Fisher Discriminant Analysis based multi-view acoustic feature reduction for physical load prediction," in *Proceedings of Interspeech*, Singapore, Sep 2014, pp. 442–446.

[24] O. Räsänen and J. Pohjalainen, "Random subset feature selection in automatic recognition of developmental disorders, affective states, and level of conflict from speech," in *Proceedings of Interspeech*, Lyon, France, Sep 2013, pp. 210–214.

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 1–27, 2011.

[26] J. C. Borda, "Mémoire sur les élections au scrutin." *Histoire de l'Académie Royale des Sciences (Académie Royale des Sciences, Paris)*, 1781.

[27] S. Chu, H. Tang, and T. Huang, "Fishervoice and semi-supervised speaker clustering," in *Proceedings of ICASSP*, Taipei, Taiwan, Apr 2009, pp. 4089–4092.

[28] B. Schuller, S. Steidl, A. Batliner, J. Epps, F. Eyben, F. Ringeval, E. Marchi, and Y. Zhang, "The INTERSPEECH 2014 computational paralinguistics challenge: Cognitive & physical load," in *Proceedings of Interspeech*, 2014.

[29] M. van Segbroeck, R. Travadi, C. Vaz, J. Kim, M. P. Black, A. Potamianos, and S. S. Narayanan, "Classification of cognitive load from speech using an i-vector framework," in *Proceedings of Interspeech*, Singapore, Sep 2014, pp. 671–675.

[30] H. Steinhaus, "Sur la division des corps matériels en parties," *Bull. Acad. Polon. Sci.*, vol. 4, no. 12, pp. 801–804, 1957.

[31] J. A. Hartigan, *Clustering Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 1975.

[32] C. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[33] S. C. Todd, M. T. Tóth, and R. Busa-Fekete, "A matlab program for cluster analysis using graph theory," *Computers & Geosciences*, vol. 36, no. 6, pp. 1205–1213, 2009.