Impacts of multicollinearity on CAPT modalities: An heterogeneous machine learning framework for computer-assisted French phoneme pronunciation training

Yanjing Bi, Chao Li, Yannick Benezeth, Fan Yang

PLOS ONE

RESEARCH ARTICLE

Impacts of multicollinearity on CAPT modalities: An heterogeneous machine learning framework for computer-assisted French phoneme pronunciation training

Yanjing Bi[1], **Chao Li**[2,3]*, **Yannick Benezeth**[4], **Fan Yang**[4]

1 School of Foreign Studies, Capital University of Economics and Business, Beijing, China, **2** Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, **3** University of Chinese Academy of Sciences, Beijing, China, **4** Laboratory ImViA, Université Bourgogne Franche-Comté, Dijon, Burgundy, France

* chao.li@mail.ioa.ac.cn

Citation: Bi Y, Li C, Benezeth Y, Yang F (2021) Impacts of multicollinearity on CAPT modalities: An heterogeneous machine learning framework for computer-assisted French phoneme pronunciation training. PLoS ONE 16(10): e0257901. https://doi.org/10.1371/journal.pone.0257901

Editor: Jie Zhang, Newcastle University, UNITED KINGDOM

Received: May 7, 2021

Accepted: September 13, 2021

Published: October 18, 2021

Data Availability Statement: All relevant data are within the paper and its Supporting information files.

Funding: This work is funded by Chinese Academy of Sciences.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Phoneme pronunciations are usually considered as basic skills for learning a foreign language. Practicing the pronunciations in a computer-assisted way is helpful in a self-directed or long-distance learning environment. Recent researches indicate that machine learning is a promising method to build high-performance computer-assisted pronunciation training modalities. Many data-driven classifying models, such as support vector machines, back-propagation networks, deep neural networks and convolutional neural networks, are increasingly widely used for it. Yet, the acoustic waveforms of phoneme are essentially modulated from the base vibrations of vocal cords, and this fact somehow makes the predictors collinear, distorting the classifying models. A commonly-used solution to address this issue is to suppressing the collinearity of predictors via partial least square regressing algorithm. It allows to obtain high-quality predictor weighting results via predictor relationship analysis. However, as a linear regressor, the classifiers of this type possess very simple topology structures, constraining the universality of the regressors. For this issue, this paper presents an heterogeneous phoneme recognition framework which can further benefit the phoneme pronunciation diagnostic tasks by combining the partial least square with support vector machines. A French phoneme data set containing 4830 samples is established for the evaluation experiments. The experiments of this paper demonstrates that the new method improves the accuracy performance of the phoneme classifiers by 0.21 − 8.47% comparing to state-of-the-arts with different data training data density.

1 Introduction

Within linguistic theories, phonemes play a central role as units of speech perception and access codes to lexical representations, and phoneme pronunciations are usually considered as basic skills for learning a foreign language. Computer-assisted pronunciation training (CAPT)

is helpful for pronunciation practice and mispronunciation identification in a self-directed or long-distance learning environment. Conventional CAPT modalities offer verification feedbacks via automatic acoustic analysis. Because phonemes are essentially 'segment-sized' (have different sizes in time domain) and abstract (have different acoustic realisations), identifying the phonemes of speeches by using some physical model can hardly satisfy the requirements of today's CAPT applications.

Recently, machine learning (ML) techniques have made great progresses, providing new opportunities to update CAPT modalities [1–8]. From the view point of computer sciences, phoneme pronunciation diagnostics are naturally target classification tasks, so we can benefit from the advances of regression analysis methods, which address the classification issues by making data-driven predictions or decisions through building a statistical model from the recorded speech data instead of analytical functions. Piotrowska et al. [1] use the parameterized audio vector as the feature vectors to improve the automatic allophone classifier, and it is reported that this method achieves an accuracy performance of 98% in the dark and clear [l] distinguishing tasks. Almajai et al. [6] develop a Deep Learning based speaker-independent speech recognizing method, which possess better accuracy performance than the conventional methods such as linear regression and maximum likelihood linear transform in the comparative evaluations. Brocki and Marasek [8] propose a DBNN-BLSTM hybrid acoustic model for large vocabulary continuous speech recognitions by combining the deep belief neural network with bidirectional long-short time memory (BLSTM) hybrid. This new method improve the recognition rate by 5% comparing to the classical BLSTM method in the low-corpus-size speech recognition tasks. Abdel-Hamid et al. [4] improve the convolutional neural network (CNN) via limited-weight-sharing scheme and use it to speech recognitions. Experiments show that it reduces the the error rate by 6–10% compares with conventional deep neural networks (DNNs) on the TIMIT phone recognition and the voice search large vocabulary speech recognition tasks. Zehra et al. [9] experimentally investigate the ensemble learning effect using a majority voting technique for cross-corpus, multi-lingual speech emotion recognition system, proving that this approach gives promising results against other state-of-the-art techniques.

However, the models of this type are sensitive to mullticollinearity of the predictors, always resulting in model distortions [10]. The multicollinearity problem means that one of the predictor variables in a classification model can be linearly predicted from the others with a substantial degree of accuracy. A set of variables is perfectly collinear if one or more exact linear relationships exists among some of the variables:

$$x_0 + q_1 x_1 + q_2 x_2 + \cdots + q_i x_i = 0 \tag{1}$$

where q_i is constant corresponding to the i-th predictor x_i. Although it is usually difficult to figure out a precise mathematical model to explain the fundamentals in a certain pattern recognition problem, many researches indicate that suppressing the multicollinearity by using some suitable method is helpful to improve the pattern discriminability. For example, Nguyen and Rocke [11] adapt partial least squares to reduce the sample vector dimensions in the analysis procedure of human tumor sample classifications based on microarray gene expressions. Uzair et al. [12] develop a hyperspectral face recognition application in the biometric field, which effectively improve the test accuracy by modeling the relations between training and prediction matrices. Li et al. [13] incorporate multicollinearity suppressing cycle into the multi-spectrum palmprint recognition framework and achieve a very high recognition rate nearly 100%.

Similarly, phoneme utterance diagnoses may also face the issue of multicollinearity problem, which is never explored in the field of CAPT. Because the utterances are made from base vibrations of vocal cords through resonance chambers (buccal, nasal and pharyngeal cavities) [14, 15], the predictors of the phoneme feature vectors are highly probably collinear. Our quantitative diagnose results demonstrate the multicollinearity problem of utterances by using condition indices (CIs) [16]:

$$CI_i = \sqrt{\frac{\lambda_{max}}{\lambda_i}} \tag{2}$$

where λ_{max} is the maximum eigenvalue of the symbol vector, and λ_i is its i-th eigenvalue. Belsey et al. [17] suggest that predictor dependencies start to affect the regression estimates when the CI is higher than 10. Fig 1 plots the condition indices of a phoneme frequency spectrum set, in which 87.27% of the elements exceed this suggested threshold line.

The work of this paper focuses on the French CAPT. It is motivated by the fact that the existing research findings demonstrate that ML-based CPAT modalities are usually distorted by the predictor collinearity. We therefore attempt to improve their accuracy performance by mitigating this problem. To do this, with the help of 23 volunteers, a new phoneme database, namely CUEB French Phoneme Database 1.0, is first established. It contains 35 phonemes ×6 sessions ×23 persons = 4830 samples, allowing us to verify the relevant theories or hypothesis. Next, as shown in Fig 1, the multicollinearity of the French phoneme utterances is analyzed, and the results indicate that it indeed exists in the case of this paper and plays a role of negative influencing factor. Thirdly, according to a state-of-art review, the partial least square (PLS) regression algorithm is used to suppress the multicollinearity of utterance sample vectors. The evaluation results show that it is an effective solution for this issue, but it is also found that the accuracy of the PLS-only classifies are lower than the typical machine learning models, i.e. support vector machines (SVMs) and DNNs. Hence, we incorporate the improved soft-margin SVMs into the target CAPT modality in order to further enhance its feature recognition ability. Finally, an heterogeneous ML framework for French phoneme pronunciation recognition is implemented and evaluated by comparing with four state-of-the-arts: PLS-only regressors, hard-margin SVM, soft-margin SVM and DNN. The innovations of this work include:

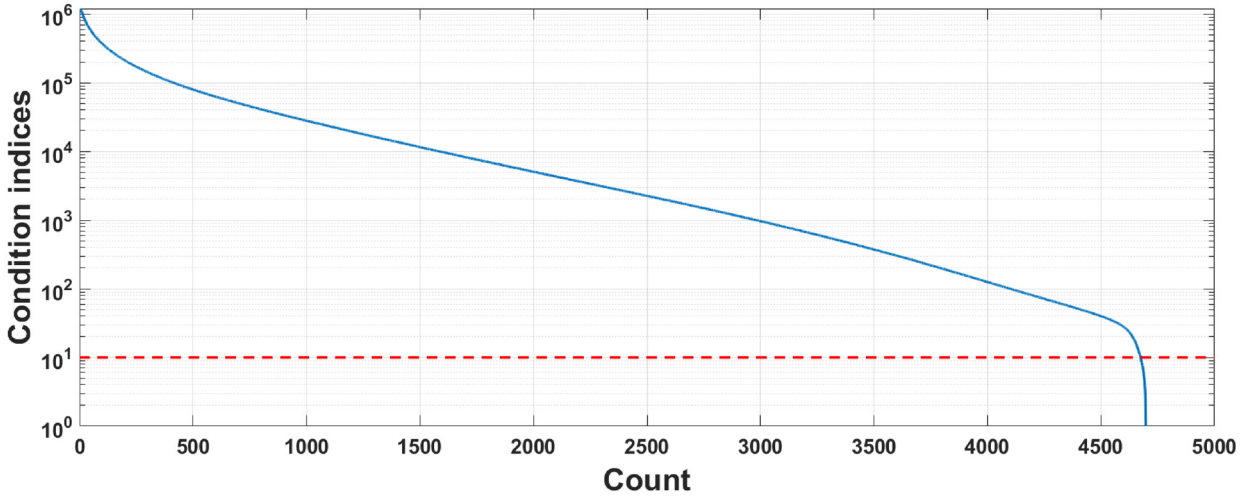


Fig 1. Condition indices of phoneme frequency spectral.

https://doi.org/10.1371/journal.pone.0257901.g001

1. The multicollinearity problem of the phoneme utterances are quantitatively analyzed and experimentally verified. Up to our knowledge, this is the first time that the phoneme utterance diagnostic problem is investigated from the view point of this theory.
2. A new heterogeneous ML framework for French phoneme pronunciation recognition is proposed. More precisely, the PLS regression algorithm is first used on the frequency spectrum of phoneme waveforms in order to suppress their multicollinearity, then the exacted features are classified via improved soft-margin SVMs. As a result, the accuracy performance of the target French CAPT modality is improved by 0.21 − 8.47% comparing to the state-of-the-arts with different data training data density.
3. A new CUEB French Phoneme Database is established. This database contains thousands of high-quality French phoneme utterance samples collected from 23 French teachers and learners, so can be used as a nice test bench in the future works.

The remainder of this paper is organized as follows: Section 2 describes the proposed CAPT framewrk; Section 3 presents the training process of the phoneme classification model; Section 4 analyzes the evaluation experiment results; finally, a conclusion is given in Section 5.

2 Proposed CAPT framework

Fig 2 shows the overall framework of the proposed CAPT. Users utter the phoneme to learn and record it as the input of the system. The input utterance is first filtered via a band-pass filter for denoising. Fig 3(a) plots a waveform example of vowel [ɑ]. Next, the waveform is segmented as follows:

$$tic = \begin{cases} t & \text{if } P(t) > \eta_{tic} \\ nan & \text{otherwise} \end{cases} \tag{3}$$

and

$$toc = \begin{cases} t & \text{if } P(t) < \eta_{toc} \\ nan & \text{otherwise} \end{cases} \tag{4}$$

where *tic* and *toc* are the start and end edge of the segmentation of interests, respectively. t is time, $P(t)$ is instantaneous power. η_{tic} and η_{toc} are two user-defined power threshold values. Fig 3(b) zooms in the segmenting result of the given waveform. Finally, the frequency spectrum of the segmentation of interest $\mathcal{F}$ is computed via Fourier Transform. As shown in Fig 3(c), the normalized frequency spectrum is used as the predictor vector of detectors:

$$\boldsymbol{x} = \frac{|\mathcal{F}| - \overline{\mathcal{F}}}{\Delta} \tag{5}$$

where $\overline{\mathcal{F}}$ is the mean of the vector $\mathcal{F}$, and Δ is the difference between its maximum and minimum values.

Finally, the predictor vector $\boldsymbol{x}$ is assigned to the corresponding detector depending on the user-selected phoneme for diagnosis. As shown in Fig 2(c), a single detector is trained specially for every phoneme. Fig 2(d) displays the architecture of the detector unit, and we can see that it is a 2-layer network architecture whose output y can be mathematically described as

$$y = \delta^{(2)}(h^{(2)}(\boldsymbol{\delta}^{(1)}(h^{(1)}(\boldsymbol{x}^{(1)})))) \tag{6}$$

Within Eq 6, $h^{(1)}$ and $h^{(2)}$ are the propagation functions of the first and second layers, whereas

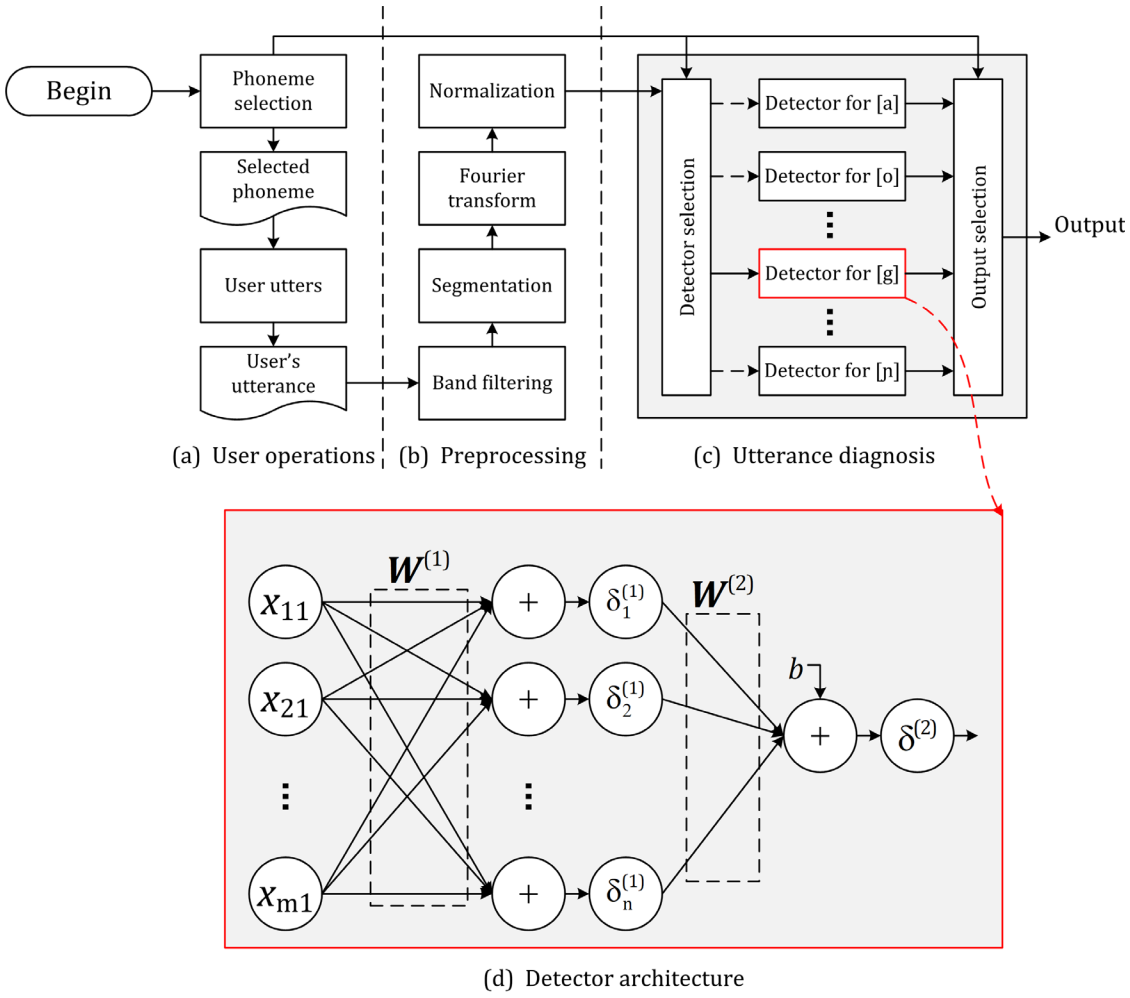
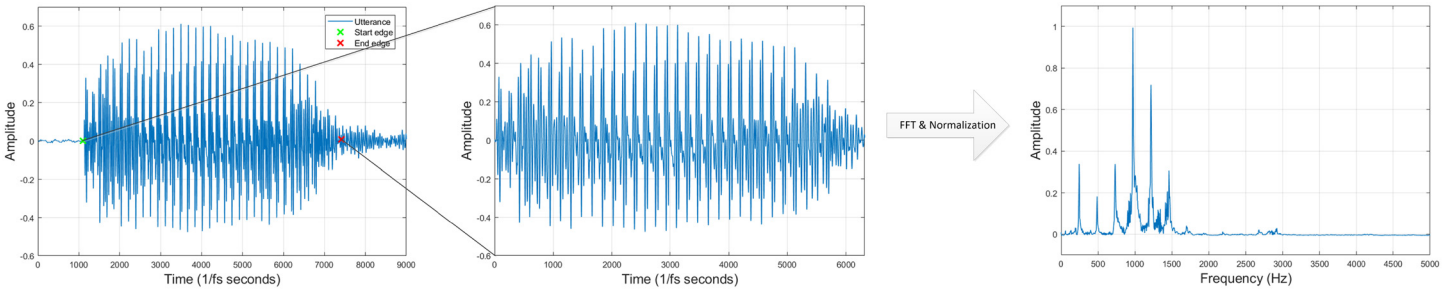


Fig 2. Proposed French CAPT framework.

https://doi.org/10.1371/journal.pone.0257901.g002

$\delta^{(1)}$ and $\delta^{(2)}$ are two activation function sets. More precisely, we have $h^{(1)}$ and $h^{(2)}$ as

$$h^{(1)}(\boldsymbol{x}^{(1)}) = \boldsymbol{x}^{(1)} \times \boldsymbol{W}^{(1)} \tag{7}$$



(a) Original waveform of utterance (b) Waveform segmentation (c) Frequency spectrum of utterance

Fig 3. Preprocessing of an utterance example of vowel [ɑ].

https://doi.org/10.1371/journal.pone.0257901.g003

and

$$h^{(2)}(\boldsymbol{x}^{(2)}) = \boldsymbol{x}^{(2)} \times \boldsymbol{W}^{(2)} + b \tag{8}$$

$\boldsymbol{x}^{(1)} = < x_{11}, x_{21}, \ldots, x_{m1} >$ (m is the vector size) is the input of the detector, so we assign the predictor vector $\boldsymbol{x}$ to it directly. $\boldsymbol{W}^{(1)}$ and $\boldsymbol{W}^{(2)}$ are the coefficient matrices of the first and second layers, respectively. Their sizes are m-by-n and n-by-1, where n is the class number of the regression task of the first layer. In the case of this paper, we set n as 35, which is the phoneme number of the French language. b is the bias value of the second layer. The training methods of the coefficient matrices and bias vectors are presented in the next section. $\boldsymbol{x}^{(2)} = < x_{12}, x_{22}, \ldots, x_{n2} >$ is the output of the first activation function set $\boldsymbol{\delta}^{(1)} =< \delta_{11}^{(1)}, \delta_{21}^{(1)}, \ldots, \delta_{n1}^{(1)} >$ and its function elements are rectified linear units (ReLU):

$$\delta_1^{(1)}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

We apply ReLUs to the elements of $h^{(1)}$'s output vector one by one. For the output of the second layer, a sigmoid function is used as its activation function in order to constrain the output of the detector into a reasonable range from 0 to 1:

$$\delta^{(2)}(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

The output of the detector y is considered as the diagnosis score, and high score values correspond to higher utterance quality. If desired, a decision can be made via a threshold η. The selected phoneme is correctly pronounced if $y > \eta$, otherwise not.

3 Training process of the utterance detectors

As shown in Fig 2(c), the utterance diagnosis is realized by using multiple independent detectors, and every detector is specified for each French phoneme. We train the detectors through an heterogeneous process combined of partial least square (PLS) regressors and soft-margin support vector machines.

3.1 First layer: PLS regression

PLS is a common class of methods for modeling relations between sets of observed variables by means of latent variables. Its underlying assumption is that the observed data is generated by a system or process that is driven by a small number of latent (not directly observed or measured) variables. Its goal is to maximize the covariance between the two parts of a paired data set even though those two parts are in different spaces. That implies that PLS regression can overcome the multicollinearity problem by modeling the relationships between the predictors. Consequently, the first layer of the detector is trained via PLS regression in order to suppress the multicollinearity among the predictors.

Let $\boldsymbol{x}_*$ be the predictor vector of a random utterance sample for training and $\boldsymbol{y}_*$ its response, where $^* = 1, 2, \ldots, N$. Both of $\boldsymbol{x}_*$ and $\boldsymbol{y}_*$ are zero-mean column vectors. We present two matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ whose i-th rows are the predictor vectors and their responses corresponding to the i-th sample. Their covariance matrix $\boldsymbol{C}_{xy}$ is given as

$$\boldsymbol{C}_{xy} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \boldsymbol{y}_i^T = \frac{1}{N}\boldsymbol{X}^T\boldsymbol{Y} \tag{11}$$

where N is the utterance sample number for training.

Next, we project the predictor vectors onto two separate directions specified by unit vectors $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ in order to obtain two random variables: $\boldsymbol{w}_x^T\boldsymbol{x}_*$ and $\boldsymbol{w}_y^T\boldsymbol{y}_*$. According to the nonlinear iterative partial least squares algorithm, PLS searches for the directions $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$ such that [12, 18]

$$\begin{aligned}\max_{\boldsymbol{w}_x,\boldsymbol{w}_y:||\boldsymbol{w}_x||=||\boldsymbol{w}_y||=1}\mathbb{C}(\boldsymbol{w}_x,\boldsymbol{w}_y) &= \max_{\boldsymbol{w}_x,\boldsymbol{w}_y:||\boldsymbol{w}_x||=||\boldsymbol{w}_y||=1}\boldsymbol{w}_x^T\boldsymbol{C}_{xy}\boldsymbol{w}_y \\ &= \max_{\boldsymbol{w}_x,\boldsymbol{w}_y:||\boldsymbol{w}_x||=||\boldsymbol{w}_y||=1}\frac{1}{m}\boldsymbol{w}_x^T\boldsymbol{X}^T\boldsymbol{Y}\boldsymbol{w}_y\end{aligned} \tag{12}$$

The directions that solve the maximal covariance optimization are the first singular vectors $\boldsymbol{w}_x = \boldsymbol{u}_1$ and $\boldsymbol{w}_y = \boldsymbol{v}_1$ of the singular value decomposition of $\boldsymbol{C}_{xy} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where the value of the covariance is given by the corresponding singular value σ_1. In this paper we apply the same data projecting strategy through deflation in order to obtain multiple projecting direction, and the deflation of $\boldsymbol{X}$ is written as

$$\boldsymbol{X}_{j+1} = \boldsymbol{X}_j(\boldsymbol{I} - \boldsymbol{u}_j\boldsymbol{p}_j^T) \tag{13}$$

with

$$\boldsymbol{p}_j = \frac{\boldsymbol{X}_j^T\boldsymbol{X}_j\boldsymbol{u}_j}{\boldsymbol{X}_j^T\boldsymbol{u}_j^T\boldsymbol{u}_j\boldsymbol{X}_j} \tag{14}$$

Let $\varphi(\boldsymbol{x}_*)$ be the feature vector of some test point. By rolling the equation above with the initialization $\varphi_1(\boldsymbol{x}_*) = \varphi(\boldsymbol{x}_*)$, a series of feature vectors in terms of the sample $\boldsymbol{x}_*$ are created:

$$\varphi_{k+1}(\boldsymbol{x}_*)^T = \varphi(\boldsymbol{x}_*)^T - \sum_{j=1}^{k}\varphi_j(\boldsymbol{x}_*)^T\boldsymbol{u}_j\boldsymbol{p}_j^T \tag{15}$$

Compute the inner products between $\varphi(\boldsymbol{x}_*)$ and $\boldsymbol{u}_i$ stored as the columns of $\tilde{\boldsymbol{U}}$:

$$\varphi_{k+1}(\boldsymbol{x})^T\tilde{\boldsymbol{U}} = \varphi(\boldsymbol{x})^T\tilde{\boldsymbol{U}} - \tilde{\varphi}(\boldsymbol{x})^T\boldsymbol{P}^T\tilde{\boldsymbol{U}} \tag{16}$$

with

$$\tilde{\varphi}(\boldsymbol{x}_*) = \varphi_j(\boldsymbol{x})^T\boldsymbol{u}_j \tag{17}$$

where $\tilde{\varphi}(\boldsymbol{x}_*)$ is the feature vector needed for the regression, and $\boldsymbol{P}$ is the matrix with the columns of $\boldsymbol{p}_j$ ($j = 1, 2, 3, \ldots, k$). For $i > j$, $(\boldsymbol{I} - \boldsymbol{u}_i\boldsymbol{p}_i^T)\boldsymbol{u}_j = \boldsymbol{u}_j$. In order to compute the regression coefficient matrix $\boldsymbol{W}^{(1)}$, we seek a coefficient matrix $\boldsymbol{B}$ that solves the following optimization [19–21]:

$$\min_{\boldsymbol{B}}||\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{B} - \boldsymbol{Y}||^2 = \min_{\boldsymbol{B}}\langle\boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{B} - \boldsymbol{Y}, \boldsymbol{X}\tilde{\boldsymbol{U}}\boldsymbol{B} - \boldsymbol{Y}\rangle \tag{18}$$

The final regression coefficients contained in $\boldsymbol{W}^{(1)}$ are given by $\tilde{\boldsymbol{U}}\boldsymbol{B}$. We solve the optimization of Eq 18 by computing its gradient with respect to $\boldsymbol{B}$:

$$\boldsymbol{B} = \frac{\sigma_j\boldsymbol{v}_j^T}{\boldsymbol{u}_j^T\boldsymbol{X}_j^T\boldsymbol{X}_j\boldsymbol{u}_j} \tag{19}$$

where $\boldsymbol{v}_j$ is the complementary singular vector associated with $\boldsymbol{u}_j$ so that

$$\sigma_j\boldsymbol{v}_j = \boldsymbol{Y}^T\boldsymbol{X}_j\boldsymbol{u}_j \tag{20}$$

It follows that the overall regression coefficients can be computed as

$$\boldsymbol{W}^{(1)} = \tilde{\boldsymbol{U}}(\boldsymbol{P}^T\tilde{\boldsymbol{U}})^{-1}\boldsymbol{C}^T \tag{21}$$

where $\boldsymbol{C}$ is the matrix with columns $\boldsymbol{c}_j = \frac{\boldsymbol{Y}^T\boldsymbol{X}_j\boldsymbol{u}_j}{\boldsymbol{u}_j\boldsymbol{X}_j^T\boldsymbol{X}_j\boldsymbol{u}_j}$.

We train the first layer of the detector by using a training set $\boldsymbol{X}$ and its responses $\boldsymbol{Y}^T$. The elements of $\boldsymbol{Y}$ are 1 if the corresponding predictor vector is matched with the user-selected phoneme and 0 otherwise. By using this way, the multicollinearity of the predictors can be well mitigated and facilitate the classifying task in the next layer.

3.2 Layer 2: Support vector machine

The second layer of the detector is trained by using improved soft-margin SVMs. SVM is a type of binary classifier that has been widely used [9, 22–24] in speech processing, and we write the SVM model desired in this paper as Eq 8. Classical SVMs build the classifier by searching for some hyperplane $(\boldsymbol{W}^{(2)}, b)$ that maximizes the margin γ between the two target clusters (correct pronunciations or not):

$$\begin{aligned} &\min_{\boldsymbol{W}^{(2)},b}\frac{1}{2}||\boldsymbol{W}^{(2)}||^2 \\ &\text{s.t.} \quad y_i(\boldsymbol{x}_i^{(2)} \times \boldsymbol{W}^{(2)} + b) \geq 1, i = 1, 2, \ldots, N \end{aligned} \tag{22}$$

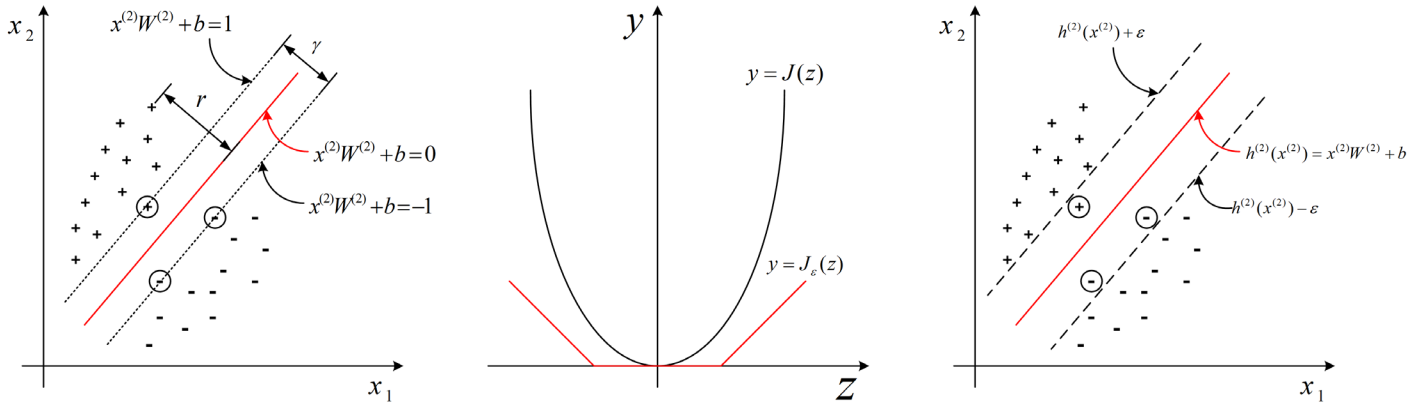
where $\boldsymbol{x}_i^{(2)}$ is the i-th predictor vector used to train the second layer, and we have

$$\boldsymbol{x}_i^{(2)} = \boldsymbol{\delta}^{(1)}(h^{(1)}(\boldsymbol{x}_i))) \tag{23}$$

Eq 22 allows to classify the utterance samples with a "hard margin" determined by support vectors (the cycled samples in Fig 4(a)), which may result in over-fitting problem. For this issue, we regularize it to

$$\min_{\boldsymbol{W}^{(2)},b}\frac{1}{2}||\boldsymbol{W}^{(2)}||^2 + C\sum_{i=1}^{N}J(h^{(2)}(\boldsymbol{x}_i^{(2)}), y_i^{(2)}) \tag{24}$$

where C is the regularization constant, and J is the loss function. The first term of Eq 24 $\frac{1}{2}||\boldsymbol{W}^{(2)}||^2$ corresponds to the structural risks, whereas the second one $C\sum_{i=1}^{N}J(h^{(2)}(\boldsymbol{x}_i^{(2)}), y_i^{(2)})$



(a) Hard-margin SVM (b) Insensitive loss function (c) Soft-margin SVM

Fig 4. Support vector regression.

https://doi.org/10.1371/journal.pone.0257901.g004

the empirical risks. This paper uses insensitive loss function J_ε as loss function:

$$J_\varepsilon(\boldsymbol{x}_i^{(2)}) = \begin{cases} 0 & \text{if } |\boldsymbol{x}_i^{(2)}| \leq \boldsymbol{\varepsilon} \\ |\boldsymbol{x}_i^{(2)}| - \boldsymbol{\varepsilon} & \text{otherwise} \end{cases} \tag{25}$$

Fig 4(b) plots the ε-insensitive loss function, and Eq 24 is rewritten as

$$\min_{\boldsymbol{W}^{(2)},b} \frac{1}{2}||\boldsymbol{W}^{(2)}||^2 + C\sum_{i=1}^{N} J_\varepsilon(h^{(2)}(\boldsymbol{x}_i^{(2)}) - y_i^{(2)}) \tag{26}$$

where $\boldsymbol{\varepsilon}$ is the maximum error between the prediction results $h^{(2)}(\boldsymbol{x}_*^{(2)})$ and the corresponding labels $y_*^{(2)}$ ($^* = 1, 2, \ldots, N$). With Eq 26, the training process takes into account the loss only when the error is higher than it. That is, a $2\boldsymbol{\varepsilon}$-width margin is obtained, within which the samples (cycled in Fig 4(c)) are supposed to have been correctly classified and their losses will not be considered.

Eq 26 is solved by using the method of lagrange multiplier. To do this, two slack variables ξ_i and ξ_i' are introduced, so that

$$\begin{aligned} \min_{\boldsymbol{W}^{(2)},b,\xi_i,\xi_i'} & \frac{1}{2}||\boldsymbol{W}^{(2)}||^2 + C\sum_{i=1}^{N}(\xi_i + \xi'_i) \\ \text{s.t.} \quad & h^{(2)}(\boldsymbol{x}_i^{(2)}) - y_i^{(2)} \leq \boldsymbol{\varepsilon} + \xi_i \\ & y_i^{(2)} - h^{(2)}(\boldsymbol{x}_i^{(2)}) \leq \boldsymbol{\varepsilon} + \xi_i' \\ & \xi_i \geq 0 \\ & \xi_i' \geq 0 \end{aligned} \tag{27}$$

with

$$i = 1, 2, \ldots, N$$

The slack variables ξ_i and ξ_i' correspond to the dissatisfaction degree to the margin constraint. We write the lagrange function of Eq 27 as

$$\begin{aligned} \mathfrak{L}(\boldsymbol{W}^{(2)}, b, \boldsymbol{\alpha}, \boldsymbol{\alpha}', \boldsymbol{\xi}, \boldsymbol{\xi}', \boldsymbol{\mu}, \boldsymbol{\mu}') &= \frac{1}{2}||\boldsymbol{W}^{(2)}||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i') - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\mu_i'\xi_i' \\ &+ \sum_{i=1}^{N}\alpha_i(h^{(2)}(\boldsymbol{x}_i^{(2)}) - y_i^{(2)} - \boldsymbol{\varepsilon} - \xi_i) \\ &+ \sum_{i=1}^{N}\alpha_i'(y_i^{(2)} - h^{(2)}(\boldsymbol{x}_i^{(2)}) - \boldsymbol{\varepsilon} - \xi_i') \end{aligned} \tag{28}$$

where $\mu_i \geq 0$, $\mu_i' \geq 0$, $\alpha_i \geq 0$ and $\alpha_i' \geq 0$, which correspond to the columns of $\boldsymbol{\mu}$, $\boldsymbol{\mu}'$, $\boldsymbol{\alpha}$, are the lagrange multipliers. Bring Eq 8 into 28 and compute its partial derivatives with respects to

$\boldsymbol{W}^{(2)}$, b, ξ_i and ξ_i':

$$\boldsymbol{W}^{(2)} = \sum_{i=1}^{N}(\alpha_i' - \alpha)\boldsymbol{x}_j^{(2)^T} \tag{29}$$

$$\sum_{i=1}^{N}(\alpha_i - \alpha_i') = 0 \tag{30}$$

$$C = \alpha_i + \mu_i \tag{31}$$

$$C = \alpha_i' + \mu_i' \tag{32}$$

According to Eqs 28–32, the dual problem of Eq 27 is obtained:

$$\begin{aligned}\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} \mathfrak{D}(\boldsymbol{\alpha},\boldsymbol{\alpha}') = \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} &\sum_{i=1}^{N}[y_i^{(2)}(\alpha_i' - \alpha_i) - \boldsymbol{\varepsilon}(\alpha_i' + \alpha_i)] \\ &-\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i' - \alpha_i)(\alpha_j' - \alpha_j)\boldsymbol{x}_i^{(2)}\boldsymbol{x}_j^{(2)^T} \\ \text{s.t.} \quad &\sum_{i=1}^{N}(\alpha_i - \alpha_i') = 0 \\ &0 \le \alpha_i, \alpha_i' \le C\end{aligned} \tag{33}$$

Within Eq 33, the lagrange multipliers α_i and α_i' correspond to the training sample $(\boldsymbol{x}_i^{(2)}, y_i^{(2)})$. For the purpose of global optima, the Karush-Kuhn-Tucker constraints must be satisfied:

$$\begin{cases}\alpha_i(h^{(2)}(\boldsymbol{x}_i^{(2)}) - y_i^{(2)} - \boldsymbol{\varepsilon} - \xi_i) = 0 \\ \alpha_i'(y_i^{(2)} - h^{(2)}(\boldsymbol{x}_i^{(2)}) - \boldsymbol{\varepsilon} - \xi_i') = 0 \\ \alpha_i\alpha_i' = 0 \\ \xi_i\xi_i' = 0 \\ (C - \alpha_i)\xi_i = 0 \\ (C - \alpha_i')\xi_i' = 0\end{cases} \tag{34}$$

The constraints of Eq 34 implies that when and only when $h^{(2)}(\boldsymbol{x}_i^{(2)}) - y_i^{(2)} - \boldsymbol{\varepsilon} - \xi_i = 0$ the value of α_i is not zero, whereas $y_i^{(2)} - h^{(2)}(\boldsymbol{x}_i^{(2)}) - \boldsymbol{\varepsilon} - \xi_i' = 0$ for that of α_i'. Additionally, it is impossible that the constraints $h^{(2)}(\boldsymbol{x}_i^{(2)}) - y_i^{(2)} - \boldsymbol{\varepsilon} - \xi_i = 0$ and $y_i^{(2)} - h^{(2)}(\boldsymbol{x}_i^{(2)}) - \boldsymbol{\varepsilon} - \xi_i' = 0$ are both valid, therefore at least one of the two multipliers α_i and α_i' have to be zero. Bringing Eq 29 into 8, we rewrite the propagation function of the second layer as

$$h^{(2)}(\boldsymbol{x}^{(2)}) = \sum_{i=1}^{N}(\alpha_i' - \alpha_i)\boldsymbol{x}^{(2)}\boldsymbol{x}_i^{(2)^T} + b \tag{35}$$

According to Eq 35 it can be seen that the predictor vectors making $(\alpha_i' - \alpha_i)$ not to be zero are

the support vectors, and they must be out of the ε-margin. Those support vectors are only parts of the training samples, so the optima of the desired SVM model is still spare.

Now we can start to compute the coefficient matrix $\boldsymbol{W}^{(2)}$ with Eqs 28–33, which is actually a quadratic programming problem and can be solved by using sequential minimal optimization (SMO) method [25]. More precisely, SMO selects one or several of them for optimizing and fixes the others so that all the variables can be solved one by one. Let α_{i_o}, α'_{i_o}, α_{j_o} and a'_{j_o} $(i_o \neq j_o)$ be the variables to be optimized for some iteration. According to the KKT constraints of Eq 34, at least one of the two multipliers α_i and α'_i have to be zero, allowing to define two of the four variables directly as zero to facilitate the optimizations. Taking $\alpha'_{i_o} = 0$ and $\alpha'_{j_o} = 0$ as an example, Eq 33 is rewritten to

$$\begin{aligned}\max_{\alpha_{i_o},\alpha_{j_o}} \mathfrak{D}(\alpha_{i_o},\alpha_{j_o},0,0) = \max_{\alpha_{i_o},\alpha_{j_o}} &\sum_{i\neq i_o}[y_i^{(2)}(\alpha'_i-\alpha_i)-\varepsilon(\alpha'_i+\alpha_i)] \\ &-\frac{1}{2}\sum_{i\neq i_o}\sum_{j\neq j_o}(\alpha'_i-\alpha_i)(\alpha'_j-\alpha_j)\boldsymbol{x}_i^{(2)}\boldsymbol{x}_j^{(2)^T} \\ &+\frac{1}{2}\alpha_{i_o}[\sum_{j\neq j_o}(\alpha'_j-\alpha_j)\boldsymbol{x}_{i_o}^{(2)}\boldsymbol{x}_j^{(2)^T}-2y_{i_o}^{(2)}-2\varepsilon] \\ &-\frac{1}{2}\alpha_{i_o}\alpha_{j_o}\boldsymbol{x}_{i_o}^{(2)}\boldsymbol{x}_{j_o}^{(2)^T} \\ \text{s.t.}\quad \alpha_{i_o}-\alpha_{j_o} = c \text{ with } c = &-\sum_{i\neq i_o,j_o}(\alpha_i-\alpha'_i) \\ 0 \leq \alpha_{i_o},\alpha_{j_o} \leq C&\end{aligned} \tag{36}$$

where the third and fourth input arguments of $\mathfrak{D}$ correspond to α'_{i_o} and α'_{j_o}, respectively. c is a constant having $\sum_{i=1}^{N}(\alpha_i-\alpha'_i)=0$ satisfied. Solve $\alpha_{i_o}-\alpha_{j_o}=c$ for a_{j_o} and substitute it in Eq 36:

$$\begin{aligned}\max_{\alpha_{i_o}} \mathfrak{D}(\alpha_{i_o},\alpha_{i_o}-c,0,0) = \max_{\boldsymbol{\alpha}_{i_o}} &\sum_{i\neq i_o}[y_i^{(2)}(\alpha'_i-\alpha_i)-\varepsilon(\alpha'_i+\alpha_i)] \\ &-\frac{1}{2}\sum_{i\neq i_o}\sum_{j\neq j_o}(\alpha'_i-\alpha_i)(\alpha'_j-\alpha_j)\boldsymbol{x}_i^{(2)}\boldsymbol{x}_j^{(2)^T} \\ &+\frac{1}{2}\alpha_{i_o}[\sum_{j\neq j_o}(\alpha'_j-\alpha_j)\boldsymbol{x}_{i_o}^{(2)}\boldsymbol{x}_j^{(2)^T}-2y_{i_o}^{(2)}-2\varepsilon+c\boldsymbol{x}_{i_o}^{(2)}\boldsymbol{x}_{j_o}^{(2)^T}] \\ &-\frac{1}{2}\alpha_{i_o}^2\boldsymbol{x}_{i_o}^{(2)}\boldsymbol{x}_{j_o}^{(2)^T} \\ \text{s.t. } \max\{0,c\} \leq \alpha_{i_o} \leq \min\{C,C+c\} \text{ with } c = &-\sum_{i\neq i_o,j_o}(\alpha_i-\alpha'_i)\end{aligned} \tag{37}$$

$\mathfrak{D}(\alpha_{i_o})$ is a quadratic polynomial in standard form, allowing to compute α_{i_o} by optimizing it within the domain [max{0, c}, min{C, C+ c}]. Similarly, the four multipliers α_{i_o}, α'_{i_o}, α_{j_o} and α'_{j_o} can be computed with the other hypothesises satisfying $\alpha_i\alpha'_i = 0$, including $\{\alpha_{i_o}=0,\alpha'_{j_o}=0\}$, $\{\alpha'_{i_o}=0,\alpha_{j_o}=0\}$ and $\{\alpha_{i_o}=0,\alpha_{j_o}=0\}$. Finally, the optimizing results with the hypothesis maximizing $\mathfrak{D}(\alpha_{i_o},\alpha_{j_o},\alpha'_{i_o},\alpha'_{j_o})$ are assigned to Eq 29 to compute the coefficient vector $\boldsymbol{W}^{(2)}$.

According to the KKT constraints of Eq 34, for every training sample $(\boldsymbol{x}_i^{(2)}, y_i^{(2)})$ it exists (C − α_i)ξ_i = 0 and $\alpha_i(h^{(2)}(\boldsymbol{x}_i^{(2)})-y_i^{(2)}-\varepsilon-\xi_i)=0$. Therefore, if the final value of α_{i_o} is neither

zero nor C, ξ_{i_o} must be zero, yielding:

$$b_{i_o} = y_{i_o}^{(2)} + \varepsilon - \sum_{j}^{N} (\alpha_j' - \alpha_j) \boldsymbol{x}_{i_o}^{(2)} \boldsymbol{x}_j^{(2)^T} \tag{38}$$

where b_{i_o} is the bias value corresponding to $(\boldsymbol{x}_{i_o}^{(2)}, y_{i_o}^{(2)})$. Theoretically, all the training samples satisfying $0 < \alpha_i < C$ should have led to the same bias, but errors may still exist due to the data distortions. For the purpose of high robustness, the values of b_i are averaged so that the final bias b is

$$b = \frac{1}{N} \sum_{i=1}^{N} b_i \tag{39}$$

with

$$b_i = \begin{cases} y_i^{(2)} + \varepsilon - \sum_{j}^{N} (\alpha_j' - \alpha_j) x_i^{(2)} x_j^{(2)^T} & \text{if } 0 < \alpha_i < C \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

4 Experiments

This section evaluates the proposed CAPT framework. The experiments are conducted by using the CUEB French Phoneme Database 1.0. First of all, the PLS regressor is tested to see whether it can mitigate the multi-collinearity of utterance waveforms. Next, the proposed method is compared with reference pronunciation diagnostic models in order to analyze its properties. All experiments have been achieved in the environment of MATLAB.

4.1 Database description

The CUEB French Phoneme Database 1.0 is established by the Capital University of Economics and Business and the Institute of Acoustics CAS. Within the Version 1.0, there are 23 participants, including 4 Chinese female teachers, 2 female French-native speakers, 3 Chinese male learners and 14 Chinese female learners. Every participants is asked to read the French phonemes shown in Table 1 six times to perform six different data sessions. The utterances are recorded at 44.1 kHz by using the private cell phones of the participants in a daily-life environment, further challenging the CAPT framework of this paper. Fig 5 plots an example of the recorded utterances. As presented in Fig 3, the sound waveform is segmented depending on the signal-to-noise ratio, and the threshold values η_{tic} and η_{toc} are $0.2 \times \bar{P}$, where $\bar{P}$ is the

Table 1. French phoneme table.

15 vowels	
Vowel:	[a], [i], [e], [ɛ], [y], [u], [o], [ɔ], [ə], [ø], [œ]
Nasal vowel:	[ã], [ɔ̃], [ɛ̃], [œ̃]
3 semi vowels	[j], [w], [ɥ]
17 consonants	
Deaf consonants:	[p], [t], [k], [f], [s], [ʃ]
Sound consonants:	[b], [d], [g], [v], [z], [ʒ]
Lateral consonants:	[l], [r]
Nasal consonants:	[m], [n], []

https://doi.org/10.1371/journal.pone.0257901.t001

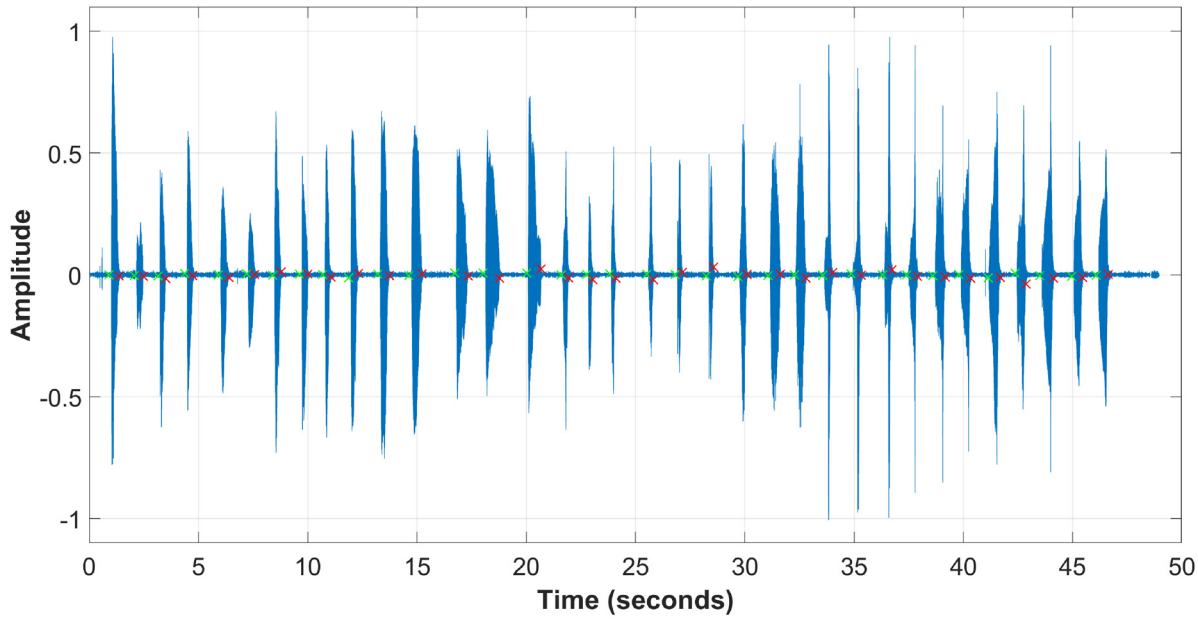


Fig 5. A data example of CUEB French Phoneme Database 1.0 (developed from S1 Audio).

https://doi.org/10.1371/journal.pone.0257901.g005

mean power of the signal. The segmentation results are marked by using green and red crosses on the waveform plot, which correspond to the start and end edges, respectively. The size of the data set of this paper is therefore 35 phonemes ×6 sessions ×23 participants = 4830 samples. Fig 6 shows the examples of the predictor vectors corresponding to the 35 French phonemes, which actually are the frequency spectrum of the utterance waveforms.

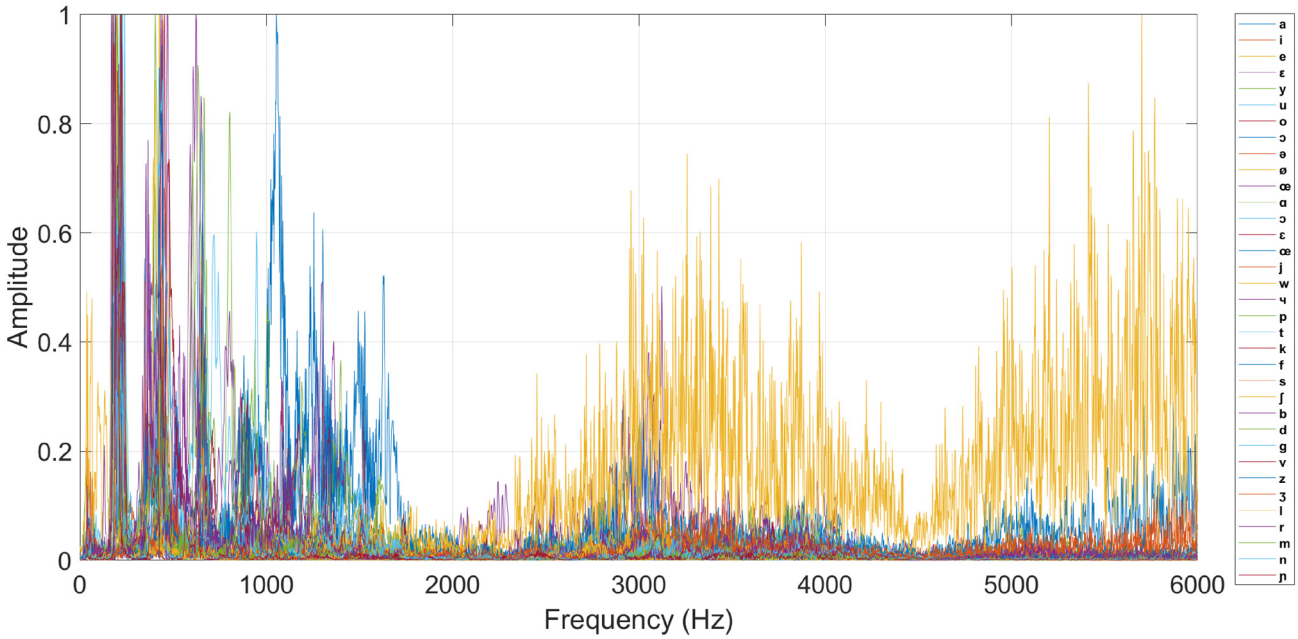


Fig 6. Examples of predictor vectors (developed from S1 Audio).

https://doi.org/10.1371/journal.pone.0257901.g006

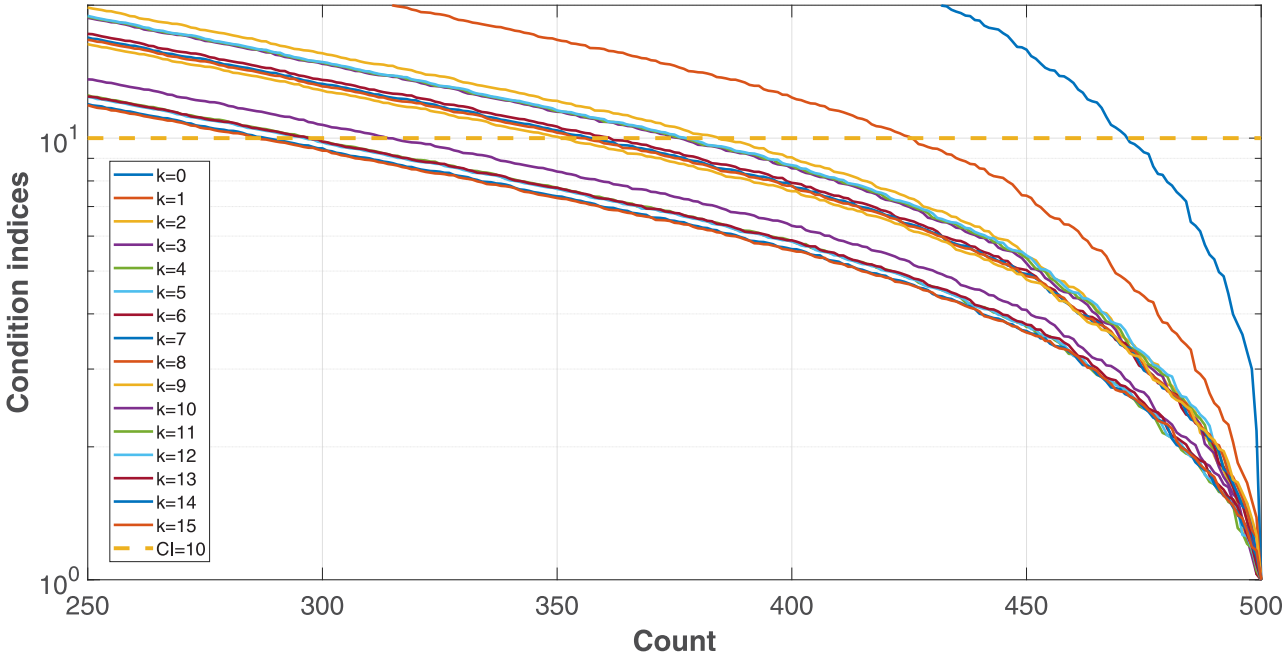


Fig 7. CI measurement results with different projection direction numbers.

https://doi.org/10.1371/journal.pone.0257901.g007

4.2 Evaluation of PLS algorithm

The subject of this experiment is to evaluate the PLS algorithm of this paper. The frequency band of the signals is from 0 to 5000 Hz, and the maximum projection direction number k of PLS regressor is 100. The dimension of the input predictor vectors is reduced down to 500 via linear interpolation. CI values are used as the criterion to quantify the multicollinearity of the processed predictor vectors (see Eq 2). All the samples of the database are used for the measurements.

Fig 7 plots the CI measurement results of the first 15 iterations, corresponding to the first 15 projection directions. It demonstrates that the CI values are reduced with the increase of the projection direction number k, indicating that the multicollinearity of the predictor vectors are mitigated step by step.

The ratios of the predictors whose CI exceed 10, which is a threshold value proposed by Belsey et al. [17] for multicollinearity estimations, are computed with different projection direction numbers. The results shown in Fig 8 demonstrate also that the multicollinearity problems are mitigated, and the high-CI predictor ratio is reduced by around 64%. Meanwhile, the proposed method loss effects when $k > 50$, implying that it possess boundary effects.

4.3 Accuracy performance

The experiments of this subsection evaluate the accuracy performance of the proposed framework as well as its sensitivity to the PLS projection direction number k. 4 of the 6 sessions of the database are used to train the framework of Fig 2 whereas the rest two for testing. The curves of receiver operating characteristics (ROC) for different user-selected phonemes are measured to estimate the minimum diagnostic error rate.

Let us take the phoneme [ɑ] for example. When this phoneme is selected, it is actually the [ɑ]-detector of Fig 2(c) who works. Fig 9 plots its ROC curves from $k = 2$ up to $k = 40$ with a

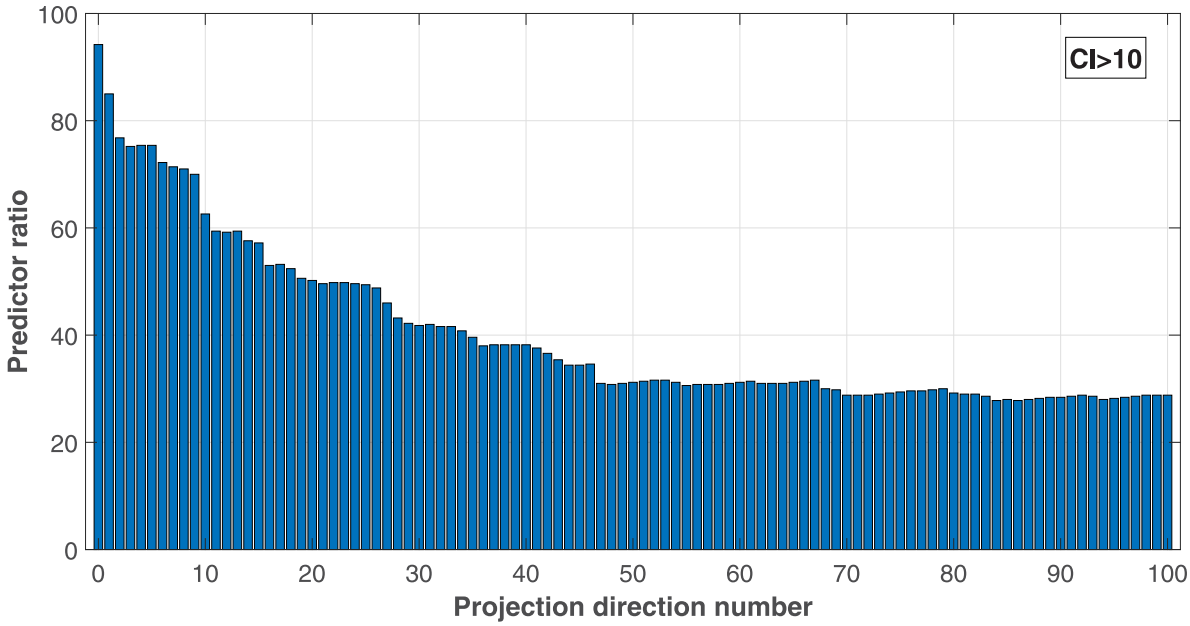


Fig 8. Predictor ratios at $CI > 10$ over the projection direction numbers.

https://doi.org/10.1371/journal.pone.0257901.g008

step of 2, in which x and y-axis correspond to the false positive rate (FPR) and false negative rate (FNR), respectively. The dotted line $FNR = FPR$ is the constraint to balance the FPRs and FNRs, meaning that the minimum diagnostic error rates are obtained when the false and leak detection rates are equal. It can be seen that with the increases of projection direction, the accuracy performance of the [ɑ]-detector is improved by around 10%, implying that PLS benefits the diagnostic tasks.

Applying the ROC measuring with the other phonemes we can get similar results. The minimum diagnostic error rates of all the phonemes over the PLS projection direction numbers are plotted in Fig 10, in which a single box corresponds to the diagnostic error rates of the 35 detectors measured with different projection direction numbers. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. We can see that the medians of the diagnostic error rates converge with the increase of projection direction number, demonstrating that the PLS algorithm helps to facilitate the classification tasks. Meanwhile, comparing to Fig 8, all the diagnostic error rate curves convergence after 8 iterations rather than 50, implying that the SVM classifier specified in this paper somehow has possessed the multicollinearity mitigating ability but cannot eliminate it completely. Fig 11 shows the optimal diagnostic error rates of all the 35 phonemes with $k \in [1, 100]$, and the overall accuracy performance of the proposed CAPT framework approximates 2.43% (average minimum diagnostic error rate).

4.4 Comparing experiments

In order to highlight the properties of the proposed method, we compare it with the state-of-the-arts. The framework shown in Fig 2 is used as the evaluation platform. Its detectors are implemented by using the classifiers to be evaluated, including PLS regressor (PLS), hard-margin SVM (HMSVM), soft-margin (SMSVM), deep neural network (DNN) and the proposed as well.

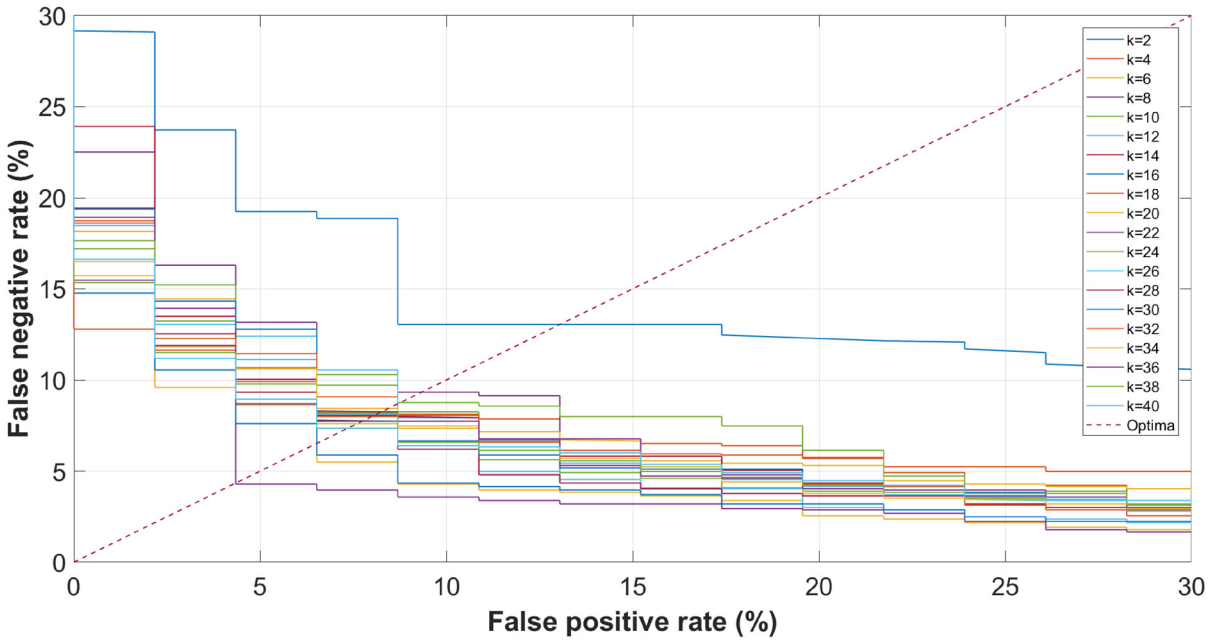


Fig 9. Receiver operating characteristics of the proposed CAPT framework with [α].

https://doi.org/10.1371/journal.pone.0257901.g009

The PLS implementation for reference is realized by using a PLS regressor which comprises of regression and classification tasks. Its final decisions are performed directly on the PLS regression without a second network layer. Its size is *m*-inputs, *m*-nodes and 1-output, where *m* is the size of the predictor vector. The HMSVM and SMSVM implementations are two

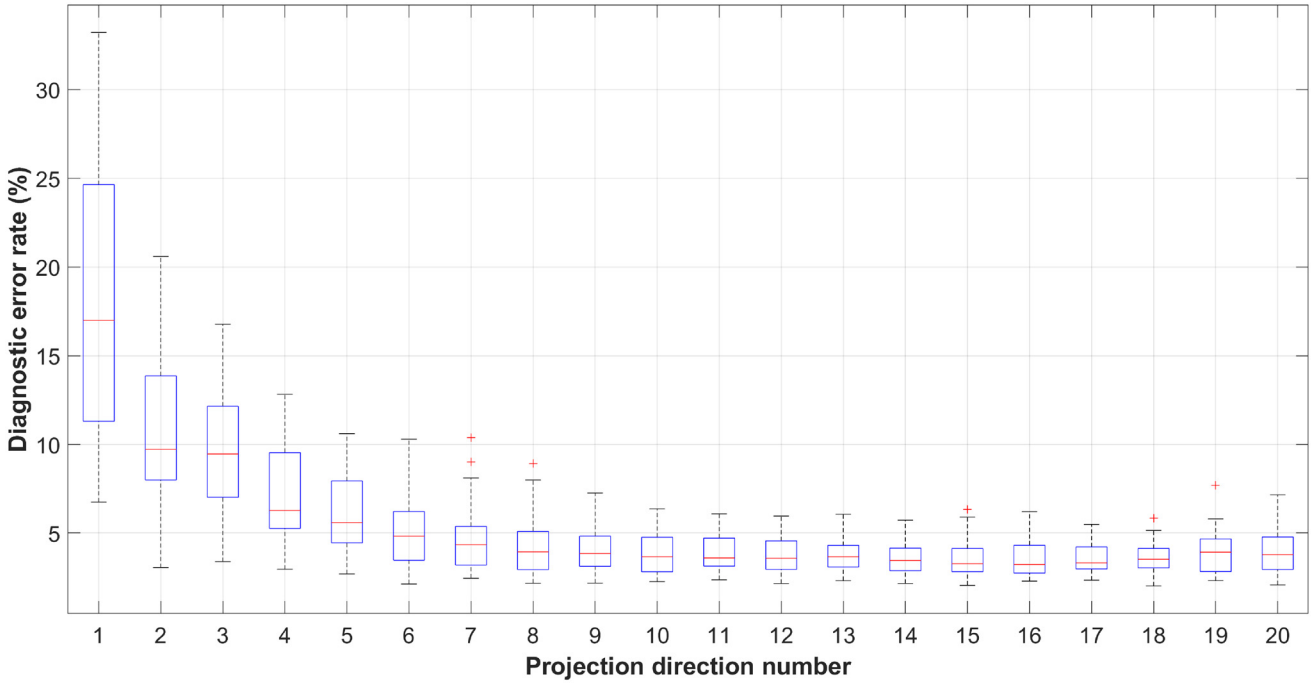


Fig 10. Diagnostic error rates with different PLS projection direction numbers.

https://doi.org/10.1371/journal.pone.0257901.g010

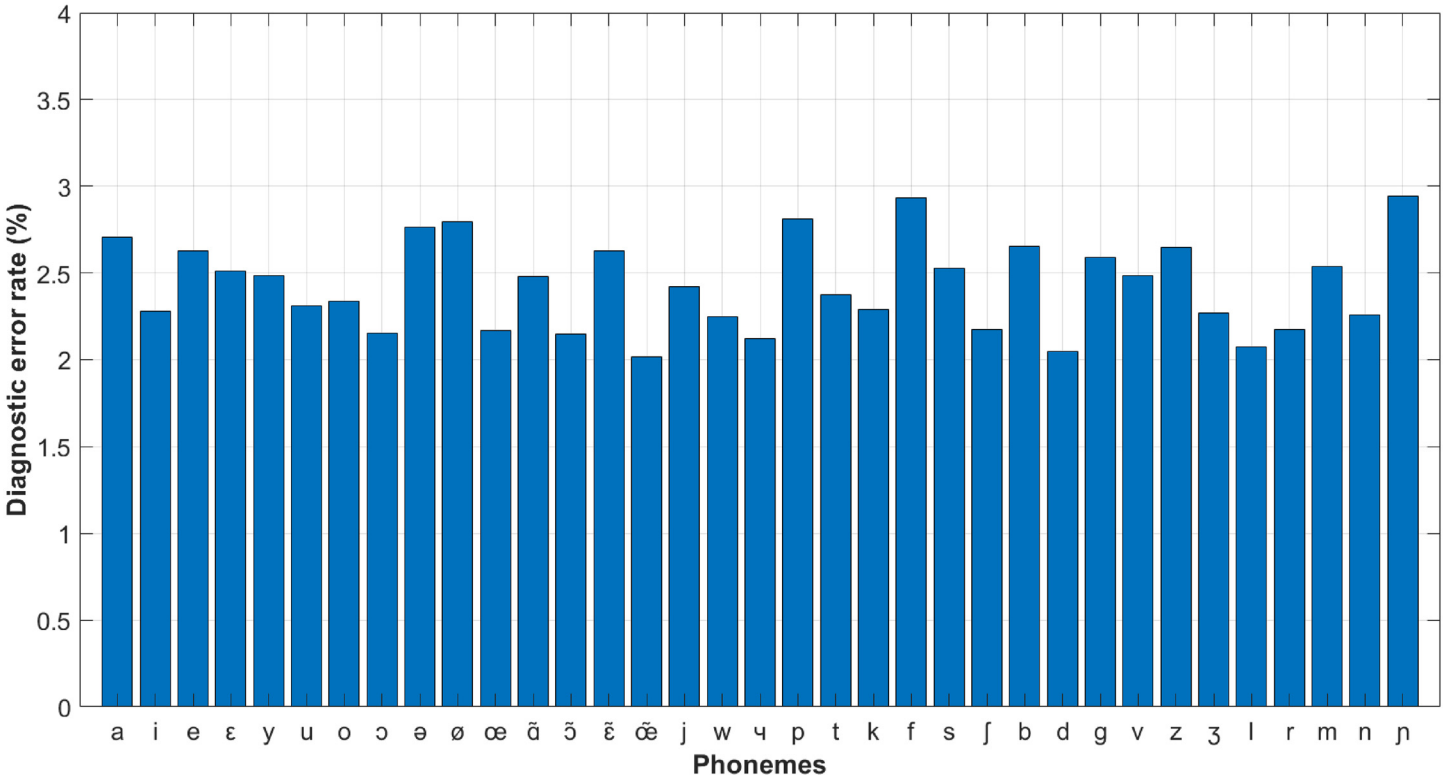


Fig 11. Minimum diagnostic error rates of all the phonemes: $k \in [1, 100]$.

https://doi.org/10.1371/journal.pone.0257901.g011

classical SMV classifiers implemented with hard and soft margins, respectively (see Fig 4). Their sizes are m-inputs, m-nodes and 1-output, their maximum training iterations are 10^6 and the minimum errors are 10^{-6}. The DNN implementation is a three-hidden-layer feedforward neural network [6]. Its size is m-inputs, $(m + 1)$-nodes per hidden layer and 1-output, the maximum training iterations is 10^6 and the minimum error is 10^{-6}.

The CAPT French Phoneme database acquires 6 sessions of data from every participant, and the experiment of this subsection divide them into two groups for training and testing randomly depending on the different ratios R = 1: 5, 2: 4, 3: 3, 4: 2 or 5: 1. For the purpose of unbiased conclusions, the average diagnostic error rate of three measurements is used as the final evaluation result. The statistical results of the evaluation are shown in Fig 12 via box-plot. A single box corresponds to the diagnostic error rates of the 35 detectors measured with different implementations and sample ratios R.

The experiment results of Fig 12 demonstrate that the diagnostic error rates of all the implementations raise with the increases of the training database size. This is because providing enough training data is a well-known solution to improve the machine learning classifiers by overcoming their over-fitting problems. The median diagnostic error rates of the five implementations reduce by 4.26%, 2.91%, 0.88%, 11.62% and 2.58%, which indicate that training data sensitivity of the proposed method is lower than PLS and DNN implementations, whereas higher than the two SVM implementations. The proposed method combines the PLS and SVM methods into a single framework, so it needs a certain number of training data to find the correct data projection directions, which raise its sensitivity to the size of the training database related to SVMs. On the other hand, the SVM layers enforce the pattern classifying ability

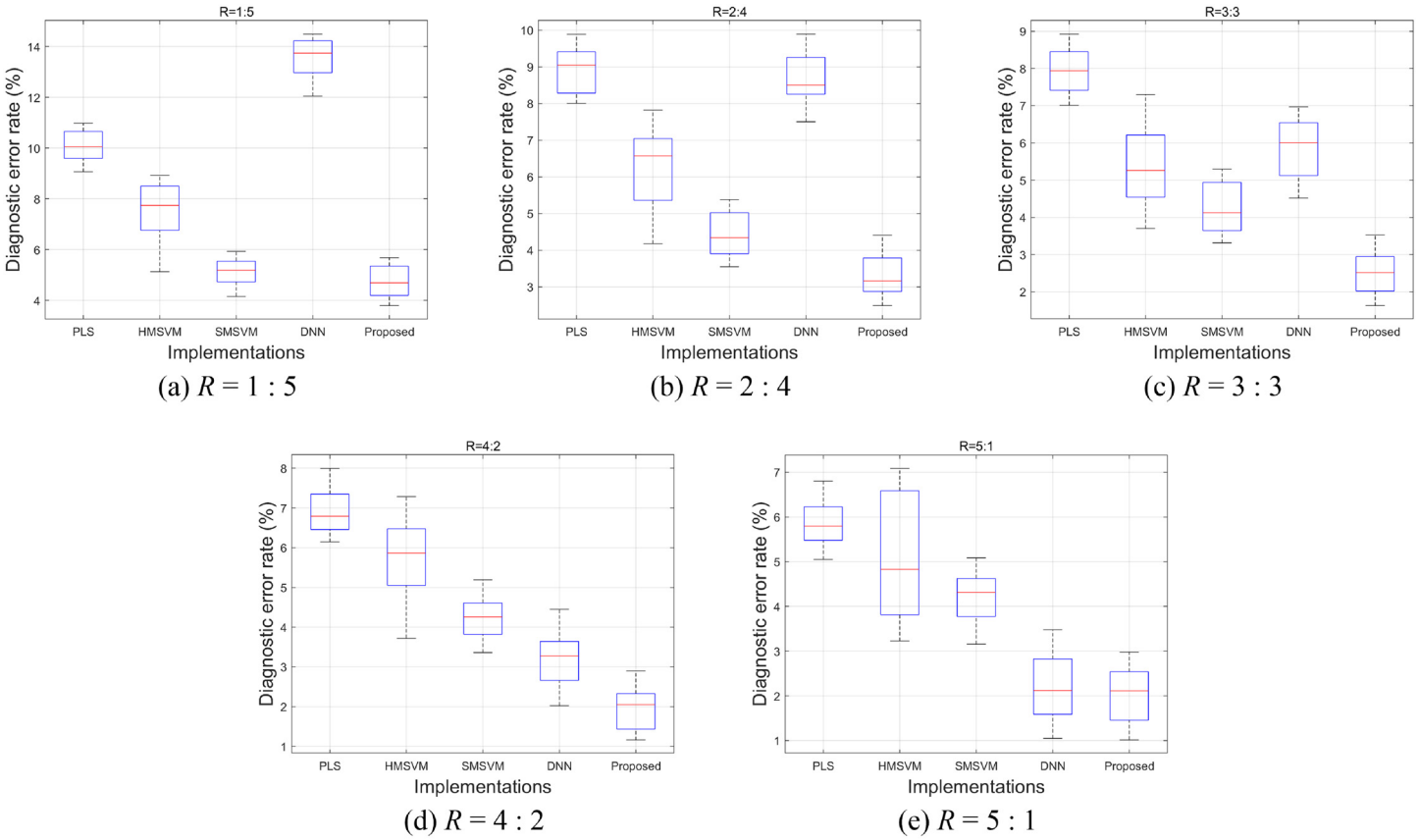


Fig 12. Diagnostic error rates of different implementations.

https://doi.org/10.1371/journal.pone.0257901.g012

of the overall CAPT framework, allowing for lower data intensity than PLS-only or DNN implementations.

Within the experiments of this paper, the proposed heterogeneous CAPT framework achieves the best performance comparing to the reference implementations. The diagnostic error rates of the implementations are further compared in Fig 13, in which each bar indicates the average diagnostic error rate of the 35 French phoneme detectors over different training-testing sample ratios. We can see that among the reference implementations, the SMSVMs achieve the best accuracy performance at $R = 1:5$, $2:4$ and $3:3$, whereas the DNNs at $R = 4:2$ and $5:1$. Comparing to them, the method of this paper improves it by 0.28%, 1.24%, 1.84%, 1.03% and 0.21%. For the proposed method itself, the accuracy achievement due to the raising of sample intensity is 2.76% ($R = 1:5$ v.s. $R = 5:1$).

5 Discussions and conclusions

This paper explores the possibility to improve the ML-based French CAPT modalities via multicollinearity suppressing. Its main contributions include:

1. The assumption that the phoneme utterance recognition models of ML families are impacted by the multicollinearity problem is experimentally verified, and a PLS based solution is proposed to address it.

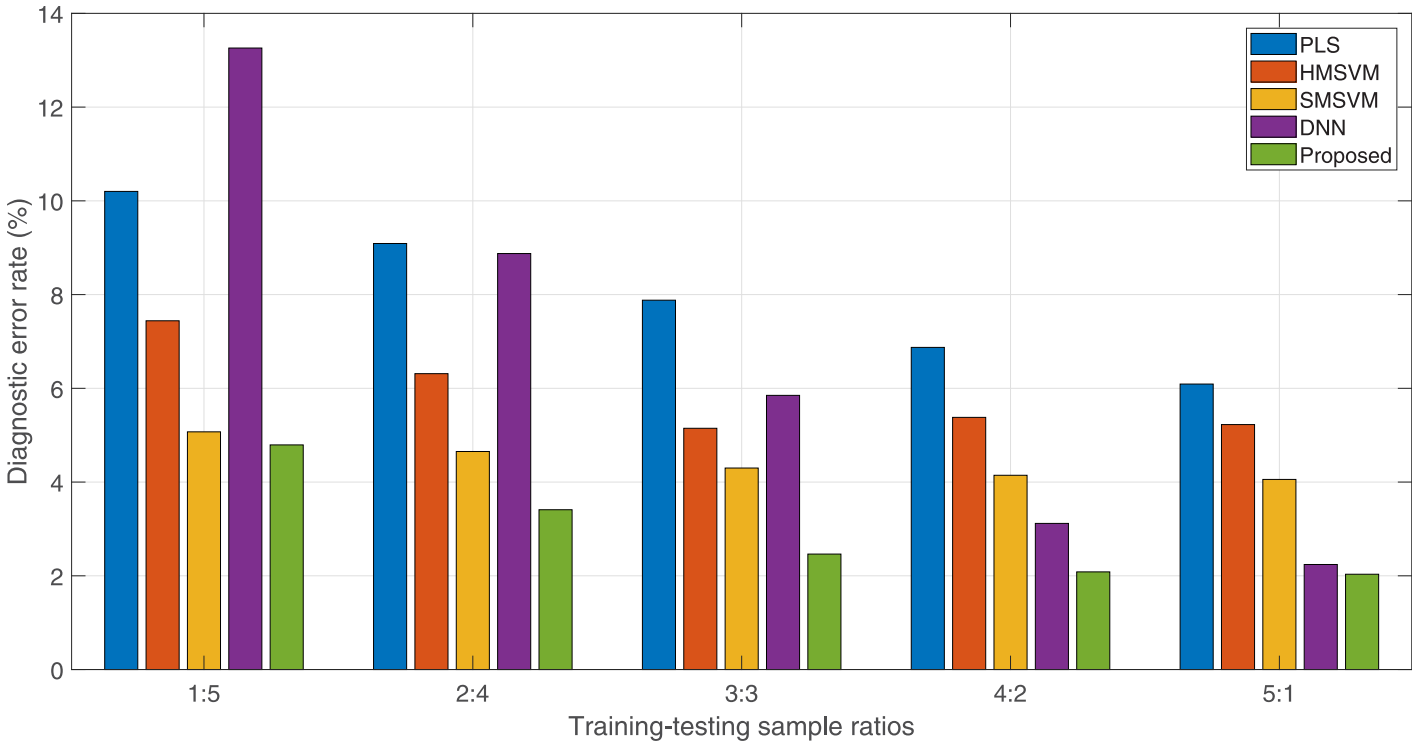


Fig 13. Average diagnostic error rates.

https://doi.org/10.1371/journal.pone.0257901.g013

2. An heterogeneous machine learning framework is developed for French CAPT modalities. It combines the PLS algorithm and improved soft-margin SVM, allowing to enhancing the classifying ability of the model by mitigating the multicollinearity problem. Evaluation results show that it achieves better accuracy performance than the reference phoneme classifying models, such as PLS-only regressor, SVMs or DNNs.
3. A French phoneme database is established in order to evaluate the achievements of this work. This database contains thousands of French phoneme utterance samples collected from 23 French teachers and learners, providing a nice test bench for future works.

Despite of achievements regarding to accuracy performance within the experiments of this paper, there exists still some issues. First of all, the proposed CAPT framework is more sensitive to the data density than SVM, but less than PLS regressor and DNN. That is, it requires more data to model the relationships between the predictors for collinearity analysis than some conventional machine learning models with low-complex topology structures. Secondly, the experiment results of Fig 13 show that the performance gap between DNN and the proposed implementations gradually shrinks with the increases of data density, implying that the multi-layer networks perhaps can handle the collinearity problem under the supports of big data as well. With the constrains of data base size, we provisionally cannot make a conclusion that DNNs will lead to better performance if enough training data are provided. Conservatively speaking, the advantage of the proposed method is to allow faster convergence with sparse training data set comparing to deep learning. Therefore, the method of this paper may be more appropriate for the scenarios of data scarcity. Finally, training a classifier of this paper is time- and resource-costly, because the PLS regressing procedure is programmatically a

dependent loop with low parallelism. At present, it seems hardly to embeddly realize such a model in a on-line way.

In the future work, we will attempt to further explore the collinearity-sensitivity characteristics of other ML classifiers, especially the methods of the deep learning families. PLS actually can be considered as a potential sparse-learning solution to address the data-hungry problem, which may better benefit the CAPT applications from deep learning methods.

Supporting information

S1 Audio. Phoneme pronunciation data example. An sample example of CUEB French Phoneme Database 1.0 conducting the experiments of this paper.
(M4A)

S1 Appendix.
(PDF)

Author Contributions

Conceptualization: Yanjing Bi, Chao Li.

Data curation: Yanjing Bi.

Funding acquisition: Yannick Benezeth.

Methodology: Yanjing Bi, Chao Li.

Project administration: Chao Li.

Software: Chao Li.

Supervision: Chao Li, Fan Yang.

Validation: Yannick Benezeth, Fan Yang.

Writing – original draft: Yanjing Bi, Chao Li.

Writing – review & editing: Chao Li, Yannick Benezeth, Fan Yang.

References

1. Piotrowska M, Korvel G, Kostek B, Ciszewski T, Cyzewski A. Machine Learning-based Analysis of English Lateral Allophones. International Journal of Applied Mathematics and Computer ence. 2019; 29 (2):393–405.
2. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521(7553):436. https://doi.org/10.1038/nature14539 PMID: 26017442
3. Schmidhuber Jürgen. Deep Learning in Neural Networks: An Overview. Neural Netw. 2015; 61:85–117. https://doi.org/10.1016/j.neunet.2014.09.003 PMID: 25462637
4. Abdel-Hamid O, Mohamed AR, Jiang H, Deng L, Penn G, Yu D. Convolutional Neural Networks for Speech Recognition. IEEE/ACM Transactions on Audio Speech & Language Processing. 2014; 22 (10):1533–1545. https://doi.org/10.1109/TASLP.2014.2339736
5. Graves A, Mohamed AR, Hinton G. Speech Recognition with Deep Recurrent Neural Networks. Acoustics Speech & Signal Processing icasspinternational Conference on. 2013.
6. Almajai I, Cox S, Harvey R, Lan Y. Improved speaker independent lip reading using speaker adaptive training and deep neural networks. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2016. p. 2722–2726.
7. Hinton G, Deng L, Yu D, Dahl GE, Mohamed Ar, Jaitly N, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. IEEE Signal Processing Magazine. 2012; 29(6):82–97. https://doi.org/10.1109/MSP.2012.2205597

8. Brocki U, Marasek K. Deep Belief Neural Networks and Bidirectional Long-Short Term Memory Hybrid for Speech Recognition. Archives of Acoustics. 2015; 40(2). https://doi.org/10.1515/aoa-2015-0021
9. Zehra W, Javed AR, Jalil Z, Gadekallu TR, Kahn HU. Cross corpus multi-lingual speech emotion recognition using ensemble learning. Complex & Intelligent Systems. 2021;(1).
10. Barons MJ, Parsons N, Griffiths F, Thorogood M. A comparison of artificial neural network, latent class analysis and logistic regression for determining which patients benefit from a cognitive behavioural approach to treatment for non-specific low back pain. In: IEEE Symposium Series on Computational Intelligence; 2013. p. 7–12.
11. Nguyen DV, Rocke DM. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics. 2002; 18(1):39. https://doi.org/10.1093/bioinformatics/18.1.39 PMID: 11836210
12. Uzair M, Mahmood A, Mian A. Hyperspectral Face Recognition With Spatiospectral Information Fusion and PLS Regression. IEEE Transactions on Image Processing. 2015; 24(3):1127–1137. https://doi.org/10.1109/TIP.2015.2393057 PMID: 25608305
13. Li C, Benezeth Y, Nakamura K, Gomez R, Yang F. A robust multispectral palmprint matching algorithm and its evaluation for FPGA applications. Journal of Systems Architecture. 2018; 88:43–53. https://doi.org/10.1016/j.sysarc.2018.05.008
14. Boersma P. An articulatory synthesizer for the simulation of consonants. In: Third European Conference on Speech Communication and Technology, EUROSPEECH 1993, Berlin, Germany, September 22-25, 1993; 1993.
15. Wong K, Lo W, Meng H. Allophonic variations in visual speech synthesis for corrective feedback in CAPT. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2011. p. 5708–5711.
16. Fildes R. Conditioning Diagnostics: Collinearity and Weak Data in Regression. Technometrics. 1993; 35(1):85–86. https://doi.org/10.1080/00401706.1993.10484997
17. David B A, Edwin K, Welsch RE. Conditioning Diagnostics: Collinearity and Weak Data in Regression. Published online: 28 january 2005 ed. Wiley-Interscience; 2005.
18. Shawe-Taylor J, Cristianini N. Kernel Methods for Pattern Analysis. New York, NY, USA: Cambridge University Press; 2004.
19. WOLD H. Soft modelling: The Basic Design and Some Extensions. Systems Under Indirect Observation, Part II. 1982; p. 36–37.
20. Wold H. In: Kotz S, Johnson NL, editors. Partial least squares. John Wiley & Sons, Inc.; 2004. Available from: http://dx.doi.org/10.1002/0471667196.ess1914.pub2.
21. Wold S, Ruhe H, Wold H, III D, J W. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverse. Journal of Scientific and Statistical Computations. 1984; 5:745–743.
22. CORTES C, VAPNIK V. SUPPORT-VECTOR NETWORKS. MACHINE LEARNING. 1995; 20(3):273–297. https://doi.org/10.1007/BF00994018
23. Schuller B, Vlasenko B, Eyben F, Wöllmer M, Stuhlsatz A, Wendemuth A, et al. Cross-Corpus Acoustic Emotion Recognition: Variances and Strategies. IEEE Transactions on Affective Computing. 2010; 1 (2):119–131. https://doi.org/10.1109/T-AFFC.2010.8
24. Albornoz EM, Milone DH. Emotion recognition in never-seen languages using a novel ensemble method with emotion profiles. IEEE Transactions on Affective Computing. 2017; 8(1):43–53. https://doi.org/10.1109/TAFFC.2015.2503757
25. Platt JC. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research; 1998. Available from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.560.