Serdica J. Computing **8** (2014), No 2, 169–182

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

ACCENT RECOGNITION FOR NOISY AUDIO SIGNALS

Zichen Ma, Ernest Fokoué

Abstract. It is well established that accent recognition can be as accurate as up to 95% when the signals are noise-free, using feature extraction techniques such as mel-frequency cepstral coefficients and binary classifiers such as discriminant analysis, support vector machine and k-nearest neighbors. In this paper, we demonstrate that the predictive performance can be reduced by as much as 15% when the signals are noisy. Specifically, in this paper we perturb the signals with different levels of white noise, and as the noise become stronger, the out-of-sample predictive performance deteriorates from 95% to 80%, although the in-sample prediction gives overly-optimistic results.

1. Introduction. Consider an audio signal $\mathbf{x}_i = (x_1, x_2, \ldots, x_p)^T$, where each vector $\mathbf{x}_i$ represents a speech signal of a speaker and the elements in the vector denote the amplitude through sampling, and $y_i \in \{1, 2, \cdots, K\}$ represents the class of accent of the corresponding speaker i. An accent recognition task is to find a classifier f that maps the signal matrix $\mathbf{X}$, where each row vector is a speech signal $\mathbf{x}_i$, onto $y \in \{1, 2, \cdots, K\}$, and that misclassification error is small.

Previous works have shown that such accent recognition tasks can be performed using some feature representation of the signals and some classifiers ([2],

ACM Computing Classification System (1998): C.3, C.5.1, H.1.2, H.2.4., G.3.

Key words: ill-posed problem, feature extraction, mel-frequency cepstral coefficients, discriminant analysis, support vector machine, k-nearest neighbors, autoregressive noise.

[20]). Also, a specific feature, the mel-frequency cepstral coefficients (MFCCs), has been shown to work well in practice ([9], [3]). It is well established in [17] that when $K = 2$, that is, a binary classification is performed, and the signals are noise-free, the prediction accuracy is as high as 95%. In this paper we demonstrate that such high performance would quickly deteriorate when the signals are contaminated by noise in the same context of binary classification. Section 2 provides a brief introduction of feature extraction with MFCCs. A review of certain pattern recognition classifiers, including the discriminant analysis, support vector machine, and the k-nearest neighbor algorithm, is provided in Section 3. Section 4 and 5 discuss the implementation of such accent recognition techniques with a designed study. In detail, Section 4 describes the study and the data, and Section 5 provides the results of predictive performance in the context of both pure signal and noisy signal. A conclusion is provided in Section 6.

2. Feature extraction using MFCCs. One problem in the tasks involving audio signals is that the dimensionality p can be readily large as a large sampling rate is used, resulting in the learning process in a large-p-small-n problem. According to [10], such large-p-small-n problem problems are ill-posed since mostly the results of such problems do not exist. Thus, it is important to find a good representation of the signals that can perform both feature extraction and dimension reduction. Here we introduce the technique of mel-frequency cepstral coefficients, which is a popular feature extraction tool in tasks such as speech recognition ([19], [12], [11], [4]) and musical instrument analysis ([16], [18]).

According to [15], the so-called MFCCs is to transform the signal from time domain to frequency domain. A paradigm of the computation of MFCCs is given in Figure 1. In this process, window functions are used due to the fact that

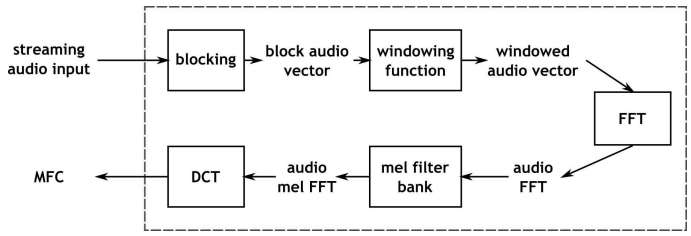

Fig. 1. A block diagram of the computation of MFCCs

Fourier transform can only be performed when the signal is stationary, which is not available when a signal is relatively long. The mel-scale is used instead of the regular hertz scale on a frequency domain in order to compensate the insensibility

of the human hearing ability at high frequency band. The relation between the two scales is given by

$$mel = \begin{cases} f & f \leq 1000 \\ 2595 \log_{10}\left(1 + \frac{f}{700}\right) & f > 1000 \end{cases} \tag{2.1}$$

It is obvious that mel-scale transforms the wide coverage at a high frequency band with hertz scale to a much narrower coverage.

In the final step of using discrete cosine transform extracting MFCCs, one can arbitrarily control the number of MFCCs to be preserved. In practice, this number is kept between 12 and 40. Also, because of the discrete cosine transform, the MFCC vectors are designed to be orthogonal to each other.

3. Techniques in binary classification.

3.1. Discriminant analysis. A standard approach to supervised classification problems is the discriminant analysis. From a Bayesian perspective, let $Y \in \{1, 2, \ldots, K\}$ be a discrete target and $\mathbf{X}$ be the data matrix. The binary classification problem can be formulated like this: *given some feature* $\mathbf{x}_i$*, classify the corresponding target* y_i *into one of the classes.* A straightforward and rather reasonable strategy is to classify y_i into the most probable class given the data ([14]). Formally, the problem can be written as

$$\hat{f}(\mathbf{x}) = \max_k Pr(Y = k \mid X = \mathbf{x}).$$

Implementing the well-known Bayes' theorem, the right-hand side can be written as

$$Pr(Y = k \mid X = \mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{f(\mathbf{x})},$$

where $f_k(\mathbf{x})$ is the conditional density of $\mathbf{x}$ in class k, π_k the prior probability of corresponding class k, and $f(\mathbf{x})$. Thus the discriminant analysis is a likelihood-based technique. That is, in order to compute the posterior probability, it is necessary to have some knowledge of the distribution of $f_k(\mathbf{x})$.

Assigning Gaussian distributions to $f_k(\mathbf{x})$ with mean μ_k and covariance matrix $\mathbf{\Sigma}_k$, the discriminant function can be written as a quadratic form

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\mathbf{x} - \mu_k)^T\Sigma_k^{-1}(\mathbf{x} - \mu_k) + \log(\pi_k) \tag{3.1}$$

and the decision rule is to assign $\mathbf{x}$ to class i if $\delta_i(\mathbf{x}) > \delta_j(\mathbf{x})$, that is,

$$\hat{f}_{QDA}(\mathbf{x}) = \arg\max_k \delta_k(\mathbf{x}). \tag{3.2}$$

A further assumption that specifies the same covariance matrix to all classes is sometimes applied. That is, the covariance matrices can be written as $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_K = \Sigma$. Under such conditions, Equation 3.1 can be further simplified to

$$\delta_k(\mathbf{x}) = \mathbf{x}^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log(\pi_k). \tag{3.3}$$

Equation 3.3 is usually called the linear discriminant function, since it only involves a linear term of $\mathbf{x}$. The decision rule is the same as a quadratic discriminant function.

Although in some applications the Gaussian assumption appears to be useful, it is in fact a rather arbitrary one. Some methods have been considered as better alternatives to the Gaussian assumption. For instance, flexible mixtures of Gaussian density can be fitted to the data and the discriminant analysis can be performed in terms of Gaussian mixture model ([13]). Or in a more general sense, the conditional densities can be estimated using kernel methods and the classification is performed based on kernel density estimation ([1]).

3.2. Support vector machine. Ever since its invention in [6], the support vector machine has been demonstrated as the state-of-the-art technique in binary classification. In its simplest case, in which there exists a linear decision boundary or a hyperplane that can completely separate the data of two classes, SVM deals with the question of what a best separation of the data should be. It concludes that the best separation is the solution of an optimization problem that seeks to maximize the distance between any observations and the linear boundary ([5]).

Mathematically, given a hyperplane on a p-dimensional space

$$\mathbf{w} \cdot \phi(\mathbf{x}) - b = 0, \tag{3.4}$$

where $\mathbf{w}$ is a coefficient vector, $\phi(\cdot)$ a function that transforms the data to a linearly separable case, $\mathbf{x}$ a point on that space, SVM can be formed as an optimization problem

$$\arg\min_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{subject to: } y_i(\mathbf{w} \cdot \phi(\mathbf{x_i}) - b) \geq 1.$$

The general SVM decision boundary is simply to replace the $\mathbf{x}$ vectors into a function $\phi(\mathbf{x})$.

$$\hat{f}_{SVM}(\mathbf{x}) = \text{sign}\left(\sum_i \hat{\alpha}_i y_i \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x}) + \hat{b}\right) \tag{3.5}$$

It is of extreme importance to notice that the nonlinear mapping function ϕ appears in the decision function in the sense of feature space inner product. Computationally, choosing the function ϕ can become infeasible quickly, while the well-known kernel trick should be used to avoid the explicit use of ϕ. Assume a kernel function K can be found so that $K(\mathbf{x}_i, \mathbf{x}) = \phi(\mathbf{x}_i)^{\mathrm{T}}\phi(\mathbf{x})$, the above decision boundary can be modified by replacing the inner product by the kernel function.

$$\hat{f}_{SVM}(\mathbf{x}) = \text{sign}\left(\sum_i \hat{\alpha}_i y_i K(\mathbf{x_i}, \mathbf{x}) + \hat{b}\right) \tag{3.6}$$

This provides great convenience in which it is not necessary to compute the nonlinear mapping explicitly, but only to perform it implicitly through the kernel. Some common kernel functions include the Radial Basis Function (RBF) kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2)$$

and the polynomial kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^d.$$

It is also of importance to know that there is no theoretical proof showing one kernel function is significantly better than others, so that which kernel function to use is an empirical question and is usually answered through the comparison of the prediction accuracy of SVM models using different kernels.

3.3. K-nearest neighbors. Compared to the above two techniques, the algorithm of k-nearest neighbors is more intuitive and is often considered as a lazy learning. The idea of this algorithm goes like this: given a dataset with known classes, or simply put, a training set, and some new data points with unknown classes, compare the distance of a new point and its first k nearest neighbors and assign the new point to the class that the majority of these neighbors lie within. For instance, when $k = 1$, we simply assign a new data point to the same class as its single nearest neighbor ([8]). That it is considered lazy learning is because no formal model is needed in this algorithm. The only requirements are a dataset in which the classes of observations are already known, some measurement for distance, and an integer k.

Mathematically, let $Tr = \{(\mathbf{x_i}, y_i) \mid \mathbf{x_i} \in \mathbb{R}^p, y_i \in \{1, 2, \cdots, S\}\}_{i=1}^n$ be a training set and $\mathbf{x}^*$ a new data point. The distances of $\mathbf{x}^*$ and $\mathbf{x}_i$'s are computed on the basis of some bivariate function $D(\cdot, \cdot)$ and ranked in an increasing order. Specify a set $\mathcal{V}_k(\mathbf{x}^*) = \{\mathbf{x_i} \mid D(\mathbf{x}^*, \mathbf{x_i}) \leq D_{(k)}\}$, where $D_{(k)}$ is the distance between the new point and the kth nearest neighbor. And the decision boundary can be

written as

$$\hat{f}_{kNN}(\mathbf{x}^*) = \arg \max_{j \in \{1,2,\cdots,S\}} \left\{ \frac{1}{k} \sum_{i=1}^{n} \mathcal{I}(y_i = j) \mathcal{I}(\mathbf{x_i} \in \mathcal{V}_k(\mathbf{x}^*)) \right\} \tag{3.7}$$

where $\mathcal{I}(\cdot)$ is an indicator function.

The crucial part of the k-NN algorithm is the distance function. Conventionally, the Euclidean distance

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{n} (x_{il} - x_{jl})^2}$$

or the Manhattan distance

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{n} |x_{il} - x_{jl}|$$

are commonly used in the computation. Also, in a binary classification, k is usually chosen to be an odd integer simply to avoid the tie-up situation.

A problem of this algorithm arises when the data contain a significant amount of noise. That is, if there is significant noise in the training set and there are some outliers in each class, the results of the classification using k-NN would degrade. Thus in this sense, k-NN is not a robust algorithm. Some substantial work has been done to remedy this drawback. Also, the algorithm of k-nearest neighbors suffers of the curse of dimensionality. That is, when the dimensionality increases, the predictive performance of this algorithm would drastically degenerate.

Even so, a theorem proven in 1960s ([7]) has shown the power of this algorithm. The theorem states that given the Bayes prediction risk R^*, which is the lowest prediction risk one can obtain, and the risk R that is given by the nearest neighbor algorithm, it has been proven that

$$R \leq 2R^* \tag{3.8}$$

The inequality (3.8) states that despite the simple algorithm, the prediction risk would not exceed the double of the lowest risk.

4. Study design and data description.

4.1. Study design without noise. In order to implement the automatic accent recognition machine and to examine the prediction ability of the algorithm, a study was constructed and the signal data are collected in the study. The procedure of the study follows the steps below:

- Through an internet resource, 22 different voices are chosen, of which 11 are American English and 11 are not. Of the non-American voices, there are 3 British English voices, 2 Spanish voices, 2 French voices, 2 Italian voices, and 2 German voices.

- Each voice is required to read 15 different multi-syllable English words, such as "approximation" and "beneficial". These words were sampled from a population of such words without replacement, which means that no words was assigned to two or more voices.

- A total of $15 \times 22 = 330$ soundtracks were recorded through some internal recording device with a sampling rate of 44,100 Hz.

At this stage, we would not want the signals to be contaminated by noise, so that we used the internet resource together with an internal recording device. Thus, the soundtracks only contained pure signals. A demographic summary of the soundtracks is given in Table 1.

Table 1. A demographic summary of soundtracks

Accent	Gender		
	Female	Male	Total
US	90	75	165
Non-US	90	75	165
Total	180	150	330

Notice that this study is balanced in terms of accent but imbalanced in terms of gender. Since this work only focuses on the recognition of different accents, we would ignore the gender of each voice at this point. Though each soundtrack only contain 1 single word and thus it is fairly short, with a sampling rate of 44,100 Hz, each one of the signal vectors contains more than 30,000 elements on time domain.

Based on the description above, the target, or response, of this classification problem is given by

$$y_i = \begin{cases} \textit{negative} & \text{if non-US,} \\ \textit{positive} & \text{if US,} \end{cases}$$

which defines this problem as a binary classification in which we are interested in categorizing different voices into two distinct accent classes. Also notice that the indices for the two classes may be not be the same in terms of different classifiers.

In discriminant analysis and k-nearest neighbors, the domain of the target can be assigned as $y_i \in \{0, 1\}$, while in the computation of SVM, this domain must be assigned as $y_i \in \{-1, 1\}$.

4.2. Perturbing signal with noise. Moreover, we can artificially perturb the signals with noise. By doing so, we would like to further examine the performance of the accent recognition algorithms under certain levels of noise. Using the signals in the same study, we are able to acquire the noisy sound by injecting some well-designed noise into the pure signals.

One such noise is the autoregressive model. An autoregressive model is a type of time-series model that specifies the output has a linear dependence only from its own previous values. In general, a pth-order autoregressive model is defined as

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \epsilon_t, \tag{4.1}$$

where ϕ_i's are the parameters of the model and usually takes values between -1 and 1 in order to keep the series stationary. The error term ϵ_t is white noise, which is a sequence of uncorrelated variables with 0 mean and finite variance σ^2 that controls how much randomness the process exhibits at each time period t. The simplest autoregressive models are the AR(0) model, in which the output at time t is the pure white noise $WN(0, \sigma^2)$, and the AR(1) model. And if we are to assign the error term a Gaussian distribution, Equation 4.1 can be modified to an AR(1) model

$$X_t = \phi X_{t-1} + \epsilon_t, \epsilon_t \sim \mathcal{N}(0, \sigma^2).$$

The value of ϕ controls the noise structure whereas the magnitude of the standard deviation σ controls the amplitude of the noise, that is, how strong the noise is.

5. Results and discussion.

5.1. Predictive performance with pure signal. As stated in Section 1, the features are extracted from raw signals and the classifiers are implemented on the features. In order to evaluate the predictive performances, a cross-validation of size 500 is used for each classifier. In detail, we examine the prediction ability of the classifiers with different numbers of MFCCs, varying from as small as 12 to as large as 39. The number of filters in the filter bank is chosen to be 40 in order to extract rich information from the signal. In terms of classification, we apply linear discriminant function, quadratic discriminant function, SVM with linear, RBF, and 2nd order polynomial kernels, and k-NN. Table 2 provides the performance results on a frequency domain. In each cell, the first

value represents the training accuracy, the second value (in *italic*) the mean prediction accuracy, and the third value (in parentheses) the standard deviation of the prediction accuracy.

Table 2. The predictive performance on frequency domain of the noise-free signals

# MFCCs	LDA	QDA	SVM-L	SVM-RBF	SVM-P	k-NN
	0.7606	0.8394	0.7758	0.9000	0.9788	0.9424
12	*0.7353*	*0.8112*	*0.7608*	*0.8208*	*0.8097*	*0.8548*
	(0.0362)	(0.0329)	(0.0351)	(0.0374)	(0.0364)	(0.0306)
	0.7818	0.9061	0.8152	0.9485	1.0000	0.9667
19	*0.7503*	*0.8647*	*0.7734*	*0.8507*	*0.8851*	*0.9098*
	(0.0345)	(0.0298)	(0.0336)	(0.0356)	(0.0274)	(0.0262)
	0.8636	0.9697	0.8667	0.9758	1.0000	0.9758
26	*0.8063*	*0.9224*	*0.8056*	*0.9080*	*0.9379*	*0.9398*
	(0.0322)	(0.0262)	(0.0337)	(0.0278)	(0.0227)	(0.0217)
	0.8970	0.9879	0.9212	0.9848	1.0000	0.9909
33	*0.8319*	*0.9543*	*0.8399*	*0.9352*	*0.9509*	*0.9586*
	(0.0314)	(0.0183)	(0.0333)	(0.0248)	(0.0205)	(0.0185)
	0.8970	0.9879	0.9152	0.9758	1.0000	0.9909
39	*0.8260*	*0.9383*	*0.8226*	*0.9223*	*0.9438*	*0.9605*
	(0.0332)	(0.0219)	(0.0326)	(0.0247)	(0.0216)	(0.0178)

Notice that the training performance exhibits an issue of over-fitting. That is, as the number of MFCCs increases, the training accuracy provides an over-optimistic performance. For some classifiers, such as SVM with polynomial kernel, the training accuracy achieves 100%, which is misleading. In terms of test or out-of-sample prediction performance, LDA and SVM with a linear kernel are close to each other and are both inferior than the other classifiers. K-NN demonstrates a better prediction ability, regardless of the number of MFCCs used. Also, it is of interest to see that there is a relatively big improvement from 12 MFCCs used, which simply indicates $p = 12$, to $p = 26$, and yet this improvement slows down from $p = 26$ to $p = 39$. For some classifiers, the accuracy even drops down slightly, from $p = 33$ to $p = 39$.

5.2. Predictive performance with noisy signal. Table 3 gives a comparison between the training accuracy and mean prediction accuracy of different classifiers with AR(0) noisy data. The number of MFCCs is 26. Various levels of σ are considered. Still, in each cell, the first value represents the training accuracy, the second value (in *italic*) the mean prediction accuracy and the third

value (in parentheses) gives the standard deviation of the prediction accuracies.

Table 3. A comparison of predictive performance with AR(0) noisy data

σ	LDA	QDA	SVM-L	SVM-RBF	SVM-P	k-NN
	0.8636	0.9697	0.8667	0.9818	1.0000	0.9758
0	*0.8063*	*0.9224*	*0.8056*	*0.9080*	*0.9379*	*0.9398*
	(0.0322)	(0.0241)	(0.0315)	(0.0304)	(0.0236)	(0.0204)
	0.8182	0.8848	0.8364	0.9606	1.0000	0.9485
0.001	*0.7708*	*0.8267*	*0.7604*	*0.9025*	*0.9048*	*0.9052*
	(0.0344)	(0.0314)	(0.0323)	(0.0260)	(0.0267)	(0.0270)
	0.8152	0.8636	0.8212	0.9636	1.0000	0.9455
0.005	*0.7600*	*0.7952*	*0.7505*	*0.8559*	*0.8416*	*0.8512*
	(0.0348)	(0.0357)	(0.0340)	(0.0342)	(0.0312)	(0.0343)
	0.8030	0.8879	0.8212	0.9394	1.0000	0.9303
0.010	*0.7539*	*0.7809*	*0.7581*	*0.8328*	*0.8079*	*0.8225*
	(0.0360)	(0.0352)	(0.0358)	(0.0344)	(0.0375)	(0.0337)
	0.8121	0.9000	0.8121	0.9364	1.0000	0.9061
0.015	*0.7611*	*0.7782*	*0.7548*	*0.8023*	*0.7809*	*0.8063*
	(0.0332)	(0.0342)	(0.0361)	(0.0362)	(0.0364)	(0.0346)

Figure 2 gives a corresponding graph of the mean predictive accuracy in Table 3, while Figure 3 plots the decrease of this accuracy compared to the noise-free case as σ increases.

Though the increment of noise does have an impact on the classifiers in terms of training accuracy, it is of importance to see that such impact is not as strong as the one in terms of prediction accuracy. That is, when the noise has a relatively large amplitude, the training accuracy exhibits strongly the feature of false optimism. This can be easily demonstrated by the classifier of SVM with a polynomial kernel. All training accuracies are exactly equal to 1 regardless of how much noise is contained in the sound, but the mean prediction accuracy drops from 93% to 78% as σ increases. It is obvious that the prediction performance would decrease as the noise in the sound gets stronger. Also, although the mean predictive accuracy decreases for all classifiers, it is of interest to see that classifiers like LDA and SVM with the linear kernel do not degrade as much as other classifiers.

Moreover, one may suspect that the performance may be affected by the various types of noise, at least within the class of AR(p) models. Another modification of the study is implemented by assigning AR(1) noise, instead of pure

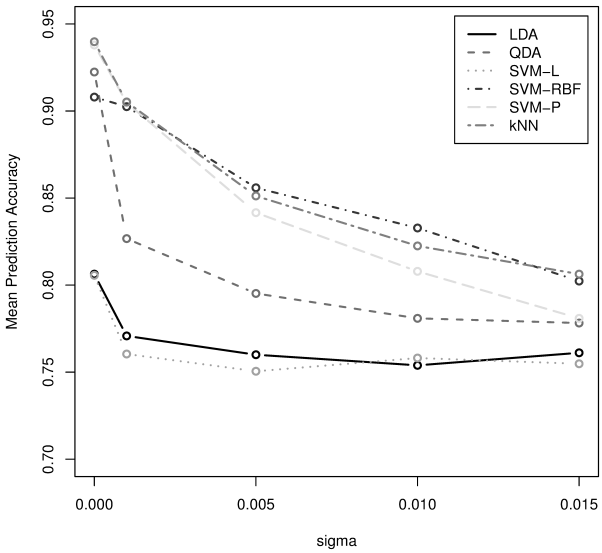

Fig. 2. A comparison of mean prediction accuracy with noisy signals

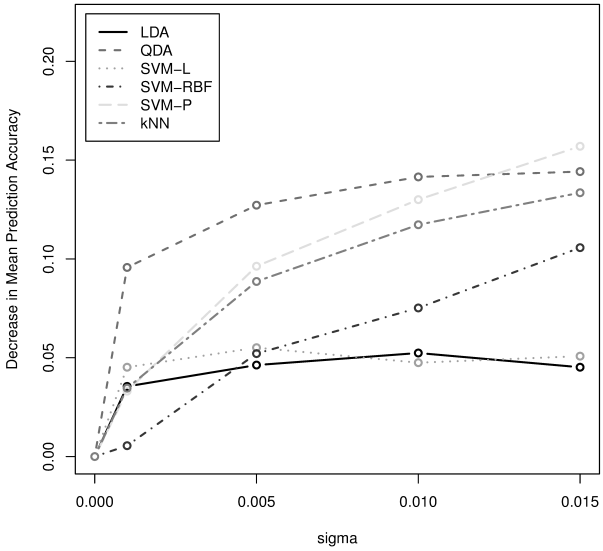

Fig. 3. Decrease mean prediction accuracy with noisy signals

white noise, to the data. The parameter ϕ is controlled at different levels between 0 and 1. Table 4 provides the result of the prediction accuracy with AR(1) noise

when the magnitude σ is fixed at 0.010. Comparing to Table 3, it seems that the

Table 4. A comparison of predictive performance with AR(1) noisy data

ϕ	LDA	QDA	SVM-L	SVM-RBF	SVM-P	k-NN
0.1	0.7642	0.7811	0.7483	0.8311	0.8037	0.8222
0.3	0.7581	0.7843	0.7480	0.8199	0.8233	0.8268
0.5	0.7604	0.7714	0.7633	0.8120	0.8054	0.8152
0.7	0.7330	0.7639	0.7512	0.8370	0.8205	0.8125
0.9	0.7424	0.7863	0.7774	0.8313	0.8104	0.8032

structure of the noise does not impact the performance, at least in terms of prediction accuracy. Also, different levels of autocorrelation do not have a significant impact to the predictive performance either.

6. Conclusion. We have demonstrated that the predictive performance of accent recognition with certain classifiers and features being extracted using MFCCs deteriorates when the signals are contaminated by noise. Comparing to the performance with noise-free signals, the accuracy drops 15% to 20% as the noise gets stronger. The possible reasons lie both on the stage of feature extraction and on pattern recognition. Though it is natural that the prediction accuracy would decrease as the data becomes noisy, such decay seems to be somewhat amplified by the technique of MFCC. In other words, the rich features being extracted from the signal do not necessarily represent the signal itself, but represent the noise instead. Such decline may also come from pattern recognition, where some classifiers are not robust to noise. Also, it is of interest to notice that the predictive performance relies more on the magnitude of the noise than its structure. Thus, in performing accent recognition with noisy signals, certain denoising techniques should be added in the step of feature extraction to at least reduce the noise to a level that cannot have a strong impact to the learning performance.

Acknowledgements. Ernest Fokoué wishes to express his heartfelt gratitude and infinite thanks to our Lady of Perpetual Help for Her ever-present support and guidance, especially for the uninterrupted flow of inspiration received through Her most powerful intercession. We both wish to thank Professor Boyan Dimitrov and Professor Leszek Gawarecki for inviting us to the Flint International Statistics Conference (FISC), we also thank the reviewers for their helpful comments and suggestions that led to the improvement of this article.

REFERENCES

[1] Baudat G., F. Anouar. Generalized Discriminant Analysis using a Kernel Approach. *Neural Computation*, **10** (2000), 2385–2404.

[2] Biadsy F. Automatic Dialect and Accent Recognition and its Application to Speech Recognition. PhD thesis submitted to Columbia University, 2011.

[3] Chechi R. Performance Analysis of MFCC And LPCC Techniques In Automatic Speech Recognition. *International Journal of Engineering Research & Technology,* **2** (2013), No 9, 3142–3146.

[4] Chen S.-H., Y.-R. Luo. Speaker Verification using MFCC and Support Vector Machine. In: Proceedings of the International Multi-Conference of Engineers and Computer Scientists, Vol. I, Hong Kong, 2009.

[5] Clarke B., E. Fokoué., H. Zhang. Principles and Theory for Data Mining and Machine Learning. Springer, NY, 2009.

[6] Cortes C., V. Vapnik. Support-vector Network. *Machine Learning*, **20** (1995), Springer, 273–297.

[7] Cover T. M. Estimation by the nearest neighbors rules. *IEEE Transactions on Information Theory*, **14** (1968), 50–55.

[8] Fokoué E. A Taxonomy of Massive Data for Optimal Predictive Machine Learning and Data Mining. Working paper CQAS-DSRG-2013-3, 2013.

[9] Gupta S., J. Jaafar. Feature Extraction using MFCC. *Signal & Image Processing: An International Journal,* **4** (2013), 101–108.

[10] Hadamard J. Lectures on Cauchy's problem in Linear Partial Differential Equations. Dover Publication, New York, 1923.

[11] Hanani A. Human and Computer Recognition of Regional Accents and Ethnic Groups from British English Speech. PhD dissertation, University of Birmingham, 2012.

[12] Hasan R., M. Jamil, G. Rabbani, S. Rahman. Speaker Identification using Mel-Frequency Cepstral Coefficients. In: Proceedings of the 3rd International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh, 2004.

[13] Hastie T., R. Tibshirani. Discriminant Analysis by Gaussian Mixtures. *Journal of the Royal Statistical Society, Series B (Methodological)*, **58** (1996), 155–176.

[14] Hastie T., J. H. Friedman, R. Tibshirani. The Elements of Statistical Learning, 2nd ed., Springer, NY, 2013.

[15] Huang X., A. Acero, H.-W. Hon. Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall, NJ, USA, 2001.

[16] Logan B. Mel-Frequency Cepstral Coefficients for Music Modelling. Cambridge Research Laboratory, 2000.

[17] Ma Z., E. Fokoué. A Comparison of Classifiers in Performing Speaker Accent Recognition Using MFCCs. *Open Journal of Statistics,* **4** (2014), 258–266.

[18] Sturm B. L., M. Morvidone, L. Daudet. Musical Instrument Identification using Multi-scale Mel-frequency Cepstral Coefficients. In: Proceedings of the 18th European Signal Processing Conference, EUSIPCO, 2010.

[19] Tivari V. MFCC and its Application in Speaker Recognition. *International Journal of Emerging Technologies,* **1** (2010), 19–22.

[20] Zheng Y. Accent Detection and Speech Recognition for Shanghai-Accented Mandarin. In: Proceedings of the EUROSPEECH-05, 2005, 217–220.

Zichen Ma
Center for Quality and Applied Statistics
Rochester Institute of Technology
98 Lomb Memorial Drive, Rochester
NY 14623, USA
e-mail: zxm7743@rit.edu

Ernest Fokoué
Center for Quality and Applied Statistics
Rochester Institute of Technology
98 Lomb Memorial Drive, Rochester
NY 14623, USA
e-mail: epfeqa@rit.edu

Received December 4, 2014
Final Accepted February 23, 2015