Exploring similarity-based classification of larynx disorders from human voice

Evaldas Vaiciukynas a,⇑, Antanas Verikas a,b, Adas Gelzinis a, Marija Bacauskiene a, Virgilijus Uloza c

a Department of Electrical & Control Equipment, Kaunas University of Technology, Studentu 50, LT-51368 Kaunas, Lithuania
b Intelligent Systems Laboratory, Halmstad University, Box 823, S 301 18 Halmstad, Sweden
c Department of Otolaryngology, Lithuanian University of Health Sciences, Eiveniu 2, LT-50009 Kaunas, Lithuania

Available online 6 May 2011

Abstract

In this paper identification of laryngeal disorders using cepstral parameters of human voice is researched. Mel-frequency cepstral coefficients (MFCCs), extracted from audio recordings of patient’s voice, are further approximated, using various strategies (sampling, averaging, and clustering by Gaussian mixture model). The effectiveness of similarity-based classification techniques in categorizing such pre-processed data into normal voice, nodular, and diffuse vocal fold lesion classes is explored and schemes to combine binary decisions of support vector machines (SVMs) are evaluated. Most practiced RBF kernel was compared to several constructed custom kernels: (i) a sequence kernel, defined over a pair of matrices, rather than over a pair of vectors and calculating the kernelized principal angle (KPA) between subspaces; (ii) a simple supervector kernel using only means of patient’s GMM; (iii) two distance kernels, specifically tailored to exploit covariance matrices of GMM and using the approximation of the Kullback–Leibler divergence from the Monte-Carlo sampling (KL-MCS), and the Kullback–Leibler divergence combined with the Earth mover’s distance (KL-EMD) as similarity metrics.

The sequence kernel and the distance kernels both outperformed the popular RBF kernel, but the difference is statistically significant only in the distance kernels case. When tested on voice recordings, collected from 410 subjects (130 normal voice, 140 diffuse, and 140 nodular vocal fold lesions), the KL-MCS kernel, using GMM with full covariance matrices, and the KL-EMD kernel, using GMM with diagonal covariance matrices, provided the best overall performance. In most cases, SVM reached higher accuracy than least squares SVM, except for common binary classification using distance kernels. The results indicate that features, modeled with GMM, and kernel methods, exploiting this information, is an interesting fusion of generative (probabilistic) and discriminative (hyperplane) models for similarity-based classification.

© 2011 Elsevier B.V. All rights reserved.

Keywords: Laryngeal disorder; Pathological voice; Mel-frequency cepstral coefficients; Sequence kernel; Kullback–Leibler divergence; Earth mover’s distance; GMM; SVM

Abbreviations: GMM – Gaussian mixture model; SVM – support vector machine; LS-SVM – least squares support vector machine; MFCC – Mel-frequency cepstral coefficient; FFT – fast Fourier transform; DCT – discrete cosine transform; RBF – radial basis function; KPA – kernelized principal angle; KGS – classical kernel Gram-Schmidt orthogonalization; MKGS – modified kernel Gram-Schmidt orthogonalization; KL – Kullback–Leibler divergence; MCS – Monte-Carlo sampling; EMD – Earth mover’s distance; MOC – minimum output coding classification scheme; SOW – single optimization by Weston & Watkins; 1v1 – one-versus-one classification scheme; 1vR – one-versus-rest classification scheme

1. Introduction

Identification of laryngeal diseases in clinical practice is a rather complex diagnostic procedure, involving evaluation of patient’s complaints, case-record, and data of instrumental as well as histological examination.
Complaints are usually summarized as questionnaire data, while the instrumental examination results into a sequence of laryngeal images and voice records. Questionnaire data and voice records are non-invasive measurements, which can be used for early detection of potential diseases and therefore be of great value in preventive care and voice quality control.

Pathological voice is induced by mass increase, a lack of closure, or elasticity change of the vocal folds. The result is that the movement of the vocal folds is not balanced and an incomplete closure of the vocal folds may appear in glottal cycles. This is the reason for changes in the whole harmonic structure (increasing the inter-harmonic energy and the fundamental frequency perturbation). Alterations related to the mucosal waveform are due to an increase of mass emerge in low bands, whereas higher bands tend to reflect noisy components due to aerial turbulence related to the mucosal waveform.

Most common classification methods, used for larynx pathology detection from voice, are metric-space classifiers, where data samples are represented as vectors of numerical features in the Euclidean space. The implicit assumption of such classifiers is that the pairwise similarity between the samples is represented by a metric distance function. For example, the distance between two samples in the Euclidean space measures their dissimilarity.

Similarity-based learning makes inferences based only on pairwise similarities or dissimilarities between a test sample and each training sample. Similarity-based learning problem can be formulated into standard learning problem in the Euclidean space by treating distances between a test sample and each training sample as features. Another popular approach, which we are using here, is to treat given similarities as inner products in some Hilbert space, that is, to modify the distances to be kernels and apply kernel classifiers (Chen et al., 2009).

However, it should be mentioned that similarity is more general than distance, since in some cases common metric distance functions fail to capture the full complexity of the relationship between the samples. The kernel trick in similarity-based classification can be applied not only upon samples, having exact coordinates in a metric space, but also upon samples in any other form. Thus the feature space in such kernel methods may not necessarily be the Euclidean one, as long as the similarity function is well defined over any pair of samples (Chen et al., 2009).

Classification methods can be categorized as generative, for example Gaussian mixture model (GMM), hidden Markov model (HMM), linear and quadratic discriminant analysis (LDA and QDA), or discriminative, such as classification and regression tree (CART), k-nearest neighbors (k-NN), multi-layer perceptron (MLP), and support vector machine (SVM). Generative models explain training data by finding optimal representation of original samples while keeping as much information as possible. Discriminative models separate data classes by finding the best decision boundary between them in some feature space. Because of this property discriminative methods often outperform generative models at classification. However, in the discriminative case we need fixed length feature vectors, while data samples in generative case can be of variable length.

Most scientific studies on laryngeal disorder detection consider simple binary classification into normal and pathological categories and choose either generative or discriminative methods. Even when methods of both types are researched, it is usually done for comparison purposes, and not for fusion sake.

An exception to this trend is the recent paper by Markaki et al., where information from different features is combined by fusing GMM and SVM classifiers (Markaki et al., 2010). In the first stage, the classification is done separately by GMM, which uses Mel-frequency cepstral coefficients as features, and SVM, employing modulation spectral features (Markaki et al., 2010). In the next stage classification results are combined. This kind of combination is of parallel nature and happens on the decision level. In our research, GMM is not used for classification, but rather helps to create a compact representation of MFCC features. Then a kernel for SVM is constructed to exploit this statistical information. Such fusion of GMM and SVM is sequential and done in the feature space.

Sequential fusion of GMM and SVM can also be achieved using GMM mean supervectors as features to SVM. The GMM mean supervector SVM provides a way of combining the generative modelling of adapted GMM mean vectors and discriminative SVM classification. GMM mean supervectors are formed through the concatenation of GMM component means, providing a convenient method of mapping a variable-length utterance to a fixed-dimension vector as required for using within an SVM classifier (McLaren et al., 2011). First use of GMM–SVM in the field of voice disorders was mentioned in Wang et al. (2011), where on the Kay Elemetrics database it was showed, that classification using GMM–SVM have superior performance than systems based on GMM, because GMM supervectors expand the features to a higher dimensional hyperplane and outperform plain GMM classification.

Even though other metric-space classifiers also use similarity or distance (dissimilarity) measures, kernel methods were chosen here as a sufficient framework to explore various similarity measures. SVM-based classification can be easily tweaked into custom similarity-based learning, just by building a proper kernel matrix.
When using conventional features for short-term signal analysis, a windowing function captures information from a short-time frame of audio. If we consider taking as many frames as possible to properly represent audio recording, then the strategies for compacting a huge and varying number of frames should be researched.

Here we chose a most popular kind of features, namely MFCCs, and compare different ways to represent the features, various schemes for multi-class classification, and SVM versus LS-SVM.

2. Voice database

Data in this study are categorized into 1 normal class and 2 classes of laryngeal disorders, namely, nodular lesions of vocal folds (nodules, polyps, and cysts) and diffuse lesions of vocal folds (papillomata, hyperplastic laryngitis with keratosis, and carcinoma). The categorization is based on visual appearance of vocal fold mass lesions from video laryngostroboscopy and direct microlaryngoscopy. Nodular lesions (localized thickenings) are single spots of various sizes with a smooth, regular surface and distinct margins surrounded by a normal tissue of the vocal fold with low malignancy potential. Respectively, diffuse vocal fold lesions are irregular, rough, multiple thickenings without distinct margins, often surrounded by an inflamed tissue, which have the potential to become cancerous. Final diagnosis was confirmed by histological examination of laryngeal specimens taken during endolaryngeal microsurgical intervention. The normal voice was recorded from randomly selected healthy volunteer individuals who considered their voice as normal. They had no complaints concerning their voice and no history of chronic laryngeal diseases or other long-lasting voice disorders. They had no symptoms concerning their voice and no history of chronic laryngeal diseases or other long-lasting voice disorders. They had no vocal complaints. The normal voice was recorded from randomly selected healthy volunteer individuals who considered their voice as normal. They had no complaints concerning their voice and no history of chronic laryngeal diseases or other long-lasting voice disorders. They had no vocal complaints.

Finally, MFCCs are obtained by applying the discrete cosine transform (DCT) to the logarithm of Mel filter bank outputs (or energies). DCT represents the signal in terms of the first basis function (constant component) and the
remaining basis functions (components of successively increasing frequency), which are uncorrelated. First 13 components of DCT represent a compacted MFCC vector of the corresponding frame. Since sometimes better results can be achieved with just 12 components (without constant component, reflecting fundamental frequency), this version of MFCC features was also tested. The Matlab code to calculate the MFCC features was adapted from the Computer Audition Toolbox (Dubnov and Yazdani, 2008), where 40 (13 linearly spaced + 27 logarithmically spaced) triangular Mel-frequency filters are used, covering the frequency range from 133 Hz to 6854 Hz. Fig. 1 illustrates the procedure to compute MFCCs.

3.2. Sampling and averaging

After converting an audio signal to cepstral coefficients, we have a vector of 13 MFCCs for each frame (window). The number of frames depends on the duration of a single recording and, in our case, ranges from 21 to 98. By cropping the audio recordings to have exact lengths, some important parts of information can be lost, so here we explain solutions how this information, residing in features, can be compressed instead of discarded. Sampling, in this context, means selecting one or several frames, i.e. to get 1 sample we select the center frame and to get more samples we select other frames, spaced evenly. Such selection of equally spaced sample frames reminds putting centered ‘comb’ on the whole recording. For example, to get 4 sample frames from 52, frame indices are calculated by these Matlab expressions:

\[
cR = \text{ceil} \left( \frac{52}{4 + 2} \right)
\]

\[
idx = \text{uint8} \left( 1 + cR + \frac{(52 - 2 \cdot cR - 1)}{(4 - 1)} \cdot [0 : 4 - 1] \right)
\]

Indices resulting from Eq. (2) are 8, 20, 33 and 45. Since voice recordings are of different length and the number of frames is not the same across them, to get a fixed number of frames (which is a pre-requisite for basic SVM-based classification without using GMM), simple time scaling can be implemented by averaging closest frames, instead of sampling. When a recording is shorter than the predefined number of frames, an extra frame is added between two neighboring frames that are closest in the Euclidean sense (have a smallest distance between their MFCC vectors). The inserted frame is the mean vector of the closest frames. When a recording is longer, such an averaged frame is placed instead of two neighboring frames. The process is repeated until the desired length is reached (recording is stretched or squeezed enough), where the number of iterations is equal to the absolute difference between the number of frames in the recording and the predefined number of frames to be left in the result (Doremalen, 2007).

Simple time-scaling, by averaging adjacent frames until the required number of them is reached, served our purposes quite well, and also outperformed the basic sampling. By using different strategies on the same data we were able to compare feature compression without GMM against the proposed GMM case, where GMMs have an attractive

---

Fig. 1. Extraction of Mel-frequency cepstral coefficients from a voice signal.
property of getting more accurate with the increasing amounts of information, i.e. the number of frames.

3.3. Gaussian mixture modeling

When applying statistical modeling, we can use a description of fixed size to represent all frames. A higher number of frames here becomes an advantage, since it results in more exact representation of statistical information. One possible solution is to represent each recording (or all recordings of single subject) with a statistical model and use it’s signature as features or to apply some parametric or non-parametric measure to assess the distance between estimated models and use the distance to calculate a kernel (Gram matrix) for classification.

Gaussian mixture modeling (GMM) can be regarded as a way of clustering and represents the data (in our case MFCCs’s) distribution as a probability density function, which is a weighted sum of K components (Gaussians):

\[ p(x) = \sum_{k=1}^{K} w_k G(x; \theta_k) \]  

where \( x \) is an \( M \)-dimensional feature vector, \( K \) is the number of components, \( w_k \) is a weight (\( w_1 + \cdots + w_k + \cdots + w_K = 1 \)), and \( G(x; \theta_k) \) is an \( M \)-variate Gaussian density with its parameters \( \theta_k \) (mean vector \( \mu_k \) and covariance matrix \( \Sigma_k \)). Given a set of feature vectors (all frames), the model parameters are estimated using an iterative expectation-maximization (EM) algorithm. Though the EM algorithm converges to a maximum likelihood it may converge to the local maximum. GMMs produced by the EM algorithm are, consequently, sensitive to initialization. Usually parameters are initialized by the \( K \)-means algorithm, which was also the case in our experiments.

Also we should note, that no generic model, like universal background model (UBM), was calculated beforehand and GMMs were created without the maximum a posteriori (MAP) adaptation. Our models, representing patients voice, don’t use tied covariance and, therefore, are not homoscedastic.

4. Classification

An SVM and its least squares simplification LS-SVM, are used for classification in this work. SVM was originally created for binary classification problems. Multi-class classification (when the number of classes \( C \geq 3 \)) usually combines several binary SVMs. The minimum output coding (MOC) requires \( L \leq C \) classifiers, where \( L = \log_2 C \). The one-vs-one (1vs1) scheme constructs a separate binary classifier for every pair of classes and yields \( C(C-1)/2 \) classifiers, while the one-vs-rest (1vsR) scheme constructs a binary classifier for each class by separating observations of this class from the rest and yields \( C \) classifiers. Decision is implemented by the voting (1vs1) or winner-takes-all strategy (1vsR). Single optimization by Weston & Watkins (SOW) attempts to directly solve a multi-class problem. This is achieved by modifying the binary class objective function and adding a constraint to it for every class (Weston and Watkins, 1999).

4.1. SVM

SVM is a large margin classifier and determines the optimal hyperplane by maximizing the margin. Some important advantages of the SVM compared to other computational intelligence techniques are good generalization properties, robustness in high dimensions, convexity of objective function, and a well-defined learning theory.

Suppose we have a set of \( N \) training samples, each represented as \((x_i, y_i)\), where \( x_i \) is the feature vector in the input space and \( y_i \) is the class label, which can be positive (+1) or negative (−1). Let \( z_i = \Phi(x_i) \) denote the corresponding feature space vector with a mapping function \( \Phi \) from the input space to a high-dimensional feature space. The hyperplane can then be defined as:

\[ w^T z + b = 0 \]  

where \( w \) is the vector defining the orientation of the hyperplane, \( b \) is the bias parameter, and \( T \) stands for transpose. Data samples are said to be linearly separable if there exists \((w, b)\), such that

\[ w^T z_i + b \geq +1 \Rightarrow y_i = +1 \]  

\[ w^T z_i + b \leq -1 \Rightarrow y_i = -1 \]

are valid for all the data samples. To deal with samples that are not linearly separable, (5) and (6) can be generalized by introducing the non-negative slack variables

\[ y_i(w^T z_i + b) \geq 1 - \xi_i \]  

where \( \xi_i \) are non-zero for those \( z_i \), which do not satisfy (5) or (6). To construct an optimal hyperplane, SVM uses an iterative training algorithm, which minimizes the error function:

\[ \frac{1}{2} ||w||^2 + C \sum_{i=1}^{N} \xi_i \]  

subject to constraints (7), where \( C \) is the capacity constant (or regularization parameter). The error function (8) is minimized by introducing Lagrange multipliers and using Kuhn–Tucker theorem of optimization theory. Non-zero coefficients in the Lagrange expansion are the so called support vectors.

4.2. LS-SVM

LS-SVM is a least squares version of the SVM, introduced by Suykens and Vandewalle (Suykens and Vandewalle, 1999).

The inequality constrains in Eq. (7) are changed to equality constrains in LS-SVM, which results in a set of linear equations. Thus, the solution to the linear system can
be calculated efficiently using a conjugate gradient method (instead of quadratic programming in SVM case)
\[
\begin{bmatrix}
0 \\
y \\
Z + y^{-1} I
\end{bmatrix} b = \begin{bmatrix}
0 \\
1
\end{bmatrix}
\]  \tag{9}
where \( Z_q = y \exp(x_i; x_j), \ I \) is the identity matrix, \( I = [I_1, \ldots, I_M], \ y = [y_1, \ldots, y_M], \) and \( x = [x_1, \ldots, x_M]. \)

Another important property of LS-SVM is that all data points become relevant and are used as support vectors. For many large scale real life applications, LS-SVM can offer a fast and simple method for obtaining classifiers with good generalization performance.

5. Kernel functions

The kernel function measures similarity or distance between a pair of variables. Once the kernel is chosen, the feature space (the subspace or space spanned by all the training samples) is automatically determined and can be used for classification.

In order to exploit GMM information, present in the covariance matrices, new kernels were created by calculating the distance between two GMMs. The similarity metrics used here were: the approximation of the Kullback–Leibler divergence from the Monte-Carlo sampling (KL-MCS) and the KL divergence combined with the Earth mover’s distance (KL-EMD). The KL-divergence approximation from the Monte-Carlo sampling is the distance measure based on the cross likelihood ratio test between samples, generated from patients’ GMM models, while the Earth mover’s distance is conceptually equivalent to the Mallows or Wasserstein distance between probability distributions.

GMM models were estimated using the Matlab toolbox Netlab, while the similarity measures (KL-MCS and KL-EMD) between them were calculated using the MA Toolbox (Pampalk, 2004). The resulting distances were further processed by the rbf_of_dist function (from the Spider toolbox (Weston and Sinz, 2006)), to get a well-formed kernel matrix \( K: \)
\[ K_{ij} = \exp \left( -\frac{D_{ij}}{2\sigma^2} \right) \]  \tag{10}
where \( D \) is a distance matrix, and \( \sigma \) is the width of the radial basis function.

For all feature extraction strategies (sampling, averaging and clustering with GMM), besides popular RBF kernel, we also explored a version of sequence kernel, based on kernelized principal angles (KPA). In the clustering case, the Gaussian mixture means (centers) alone were used as features for classification with the RBF and KPA kernels.

5.1. Kullback–Leibler divergence

The Kullback–Leibler divergence, also known as mutual information, relative entropy or, simply, information divergence, is a classical information gain measure of the asymmetric difference between two distributions, i.e. it measures the divergence from one probability distribution to another. The KL divergence is not a metric in a true sense, since it is not symmetric and does not satisfy the triangle inequality (Tversky and Gati, 1982), however, the symmetric KL-divergence between two distributions \( p \) and \( q \) (two GMMs, for example) may be expressed as
\[
D(p, q) = \int p(x) \log \frac{p(x)}{q(x)} + \int q(x) \log \frac{q(x)}{p(x)}
\]  \tag{11}
The KL divergence assumes values in the range between 0 and infinity, and is 0 if and only if the two distributions are identical.

5.2. Monte-Carlo sampling

A closed form expression for KL divergence only exists when the number of Gaussians in a mixture is 1. We can use Monte-Carlo simulations to approximate the KL-divergence between two non-single Gaussian mixtures \( p \) and \( q \) as follows:
\[
D(p, q) = \sum_{i=1}^{T} \log \frac{p(x_{pi})}{q(x_{qi})} + \sum_{j=1}^{T} \log \frac{q(x_{qi})}{p(x_{pi})}
\]  \tag{12}
where \( x_{pi} \) and \( x_{qi} \) are either the real data observations that were used to estimate the parameters of \( p \) and \( q \) or they are synthetic samples, i.e. randomly generated from the estimated probability distributions \( p \) and \( q \), and \( T \) is the number of observations or samples. The above approximation of the KL-divergence is, exactly, the distance measure based on the cross likelihood ratio test. The drawback of the Monte-Carlo approach is that even though we have a compact probabilistic representation of data in the GMM form, we still have to refer back to the original data, and because of the stochastic nature of the Monte-Carlo method, approximations could vary in different runs.

5.3. Earth mover’s distance

We use the Earth mover’s distance to calculate the distance between the probability distributions in each dimension and in such a way compute the distance between recordings of two subjects.

Instead of comparing the values of each GMM center, the minimum amount of work needed to transform one distribution into the other, or supplying points into demanding points, is calculated (Kuroiwa et al., 2006). This could be imagined as the least work required to flatten the land, where source distribution are hills and target distribution are valleys.

The computation is based on a simplified solution to the transportation problem where the total supply equals the total demand (sum of priors of source and target GMMs are equal) and can be solved by linear optimization:
The kernel (similarity metric) is given by:

$$D(p, q) = \frac{\sum_i g_{ij} d_{ij}}{\sum_i g_{ij}}$$  \hspace{1cm} (13)$$

where \(d_{ij}\) denotes the dissimilarity between GMM centers \(i\) and \(j\), and \(g_{ij}\) is the optimal flow between the two distributions such that the total cost \(\sum_i g_{ij} d_{ij}\) is minimized, subject to the following constraints:

$$\sum_i g_{ij} \leq p_i \quad \sum_j g_{ij} \leq q_j \quad \sum_{ij} g_{ij} = \text{min}(p_i, q_i)$$  \hspace{1cm} (14)$$

for all \(i\) and \(j\). \(\sum_{ij} g_{ij}\) is a normalization factor that permits matching parts of distributions with different total mass.

The EMD is conceptually equivalent to the Mallows’ or Wasserstein’s distance between probability distributions, also known as Kantorovich metric, and in the case of two distributions with equal masses, they are exactly the same (Levina and Bickel, 2001).

### 5.4. Radial basis function

A radial basis function (RBF) is by far the most popular choice of kernel types used in SVM classification. The RBF kernel function is given by

$$K(x_i, x_j) = \exp \left( -\frac{||x_i - x_j||^2}{\sigma^2} \right)$$  \hspace{1cm} (15)$$

where \(\sigma\) is the width of the basis functions.

### 5.5. Kernel principal angles

Ensemble matching methods generally consider a task of obtaining a similarity function which operates on pairs of sets of feature vectors (matrices) or ensembles.

The principal angle is the angle between two linear subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces. The degree of alignment of two subspaces of two matrices, each matrix composed of feature vectors as columns. Kernelizing this angle via the kernel trick, allows it to be calculated between non-linear subspaces.

When using this kernel, larynx pathology recognition is done on the premises, that different disorders generate different subspaces and the degree of alignment (principal angles) between them can be measured. A positive-definite kernel (similarity metric) is given by:

$$K(X_i, X_j) = \prod_{k=1}^{K} \cos^2(\varphi_k)$$  \hspace{1cm} (16)$$

where \((X_i, X_j)\) is a pair of matrices and \(\cos(\varphi_k)\) are often referred to as principal correlations or canonical correlations of the matrix pair. If the angle \(\varphi_k = 0\), then \(\cos(\varphi_k) = 1\) and the vectors are said to be parallel. If the angle \(\varphi_k = 1\), then \(\cos(\varphi_k) = 0\) and the vectors are said to be orthogonal.

The kernel trick is performed here for correlation, to compute principal angles in the feature space induced by a minor Gaussian (RBF) kernel (Wolf and Shashua, 2003). Using a linear minor kernel (polynomial degree 1) here would be equivalent to computing principal angles in the input space (between linear subspaces).

The modified version of the kernel Gram-Schmidt (MKGS) orthogonalization was used to compute principal correlations. The MKGS algorithm (Zheng, 2006) for QR decomposition in the feature space, used in this work, is much more numerically stable than the classical kernel Gram-Schmidt (KGS) version (Wolf and Shashua, 2003).

### 6. Experiments

#### 6.1. Experimental setup

To evaluate the generalization error, stratified 10-fold cross validation was used. The database of patients was split into 10 approximately equal partitions and each fold was selected in turn for testing while training on the remaining data (i.e. 9/10 for training + 1/10 for testing). When splitting the data, random sampling was done to guarantee that each class is properly represented in every fold, i.e. classes are in balanced proportions with respect to the initial distribution of the database. Finally, after all folds are tested, the average accuracy and it’s variance of the researched model can be calculated.

The test of every model results in accuracies of 10 folds from the cross validation procedure. To select the best performing model, comparison of the results was done with the help of the right tailed independent two-sample t-test. When variances of accuracy were found to be unequal, the t-test variant with the Behrens–Fisher’s problem was chosen.

Fig. 2 illustrates the training and testing phases of the proposed GMM–SVM technique. By training the SVM, a classification hyperplane is formed in the GMM space and then employed to test and score unseen samples.

The appropriate values of SVM parameters were selected experimentally by exhaustive grid search of various \(C\) and \(\sigma\) (kernel width) combinations. Values were taken on a logarithmically growing scale with 10 steps in a range of 0.1 to 100 for \(C\) and 1 to 100 for \(\sigma\), so, in total, 100 different combinations of SVM parameters were tested for each configuration. In multi-class classification, almost all classification schemes use more than one SVM, but parameters in our experiments were not tuned individually, instead the same values of \(C\) and \(\sigma\) were set for each SVM instance.

Before the classification, pre-processed MFCCs were normalized to zero mean and unit variance. For Monte-Carlo sampling the best experimentally found value was 200 samples.

#### 6.2. Results

In the figures, SOW, 1vs1 and 1vsR stand for the coding schemes of pure SVM, while successive MOC, 1vs1 and 1vsR labels denote coding schemes used in LS-SVM. The
1vs1 and 1vsR coding schemes need three classifiers, the MOC scheme uses two classifiers, and SOW solves a multi-class problem with a single classifier.

Figs. 3–6 present the classification accuracy of the best performing SVM and LS-SVM models, obtained for various kernels. Accuracy, represented in the figures, is the mean classification accuracy of all 10 folds tested. Confidence limits for the mean were calculated and also shown in the figures. The 95% confidence interval was chosen.

On average, the sequence kernel (KPA) has shown a more stable and slightly better accuracy than the Gaussian (RBF) kernel, see Figs. 3 and 4. However, the difference in accuracy was not statistically significant in most of the cases.

Considering the results presented in Fig. 3 for the averaging and sampling strategies, seems that averaging (top) tends to retain more information for classification and therefore is more useful than sampling (bottom). However, the performance gain is not statistically significant.

When using a variant of GMM supervector kernel, it was found that it is better to concatenate the means of separate GMMs (Fig. 4 (bottom)) made from each recording, rather than make a single GMM from all recordings of a subject and use its means (Fig. 4 top).

The KL-MCS kernel performed better when the full covariance matrix was used for GMM, rather than a diagonal one, see Fig. 5 (bottom). However, the KL-EMD kernel has shown the opposite behavior, see Fig. 5 (top). The KL-EMD kernel, using GMM with a diagonal covariance matrix, provided the best overall performance.

Comparing the results presented in Fig. 5 with those shown in Fig. 3 and 4 one can see that, the distance kernels (KL-MCS and KL-EMD) outperformed the RBF and KPA ones. In most cases, the difference in accuracy was statistically significant.

As can be seen from the figures, the MOC scheme performed significantly worse than the other techniques, whereas the SOW classification scheme did surprisingly well. The ordinary SVM (left-hand side of the plot) was superior to the LS-SVM (right-hand side of the plot), in the three-class classification.

Fig. 6 (bottom) shows the best results of common binary classification into normal and pathological categories.

7. Discussion and conclusions

By exploring kernels in SVM-based classification of larynx pathology from voice, we aimed at outlying that the feature space in kernel methods may not necessarily be the Euclidean one. The kernel trick, in this context, can be applied not only upon samples, having exact coordinates in a metric space, but also upon samples in any other form, as long as the similarity function is well defined over any pair of samples (Chen et al., 2009). Therefore a similarity-based paradigm has wider perspective than usual metric-space classifiers, and kernel methods could be successfully used for this kind of learning.

Notions of similarity also appear to play a fundamental role in human learning. Set-theoretic similarity models
have been successful in explaining human judgement in various similarity assessment tasks, and are consistent with the observations, made by psychologists, that metrics do not account for cognitive judgement of similarity in complex situations and that such distances may be asymmetric and may not follow the triangle inequality law (Tversky and Gati, 1982). Therefore, similarity-based models may be useful for imitating or understanding how humans learn and categorize (Chen et al., 2009).

As could be seen from the results, in all three-class classification tasks, SVM provided a higher accuracy than LS-SVM, while in common binary classification, using distance kernels, both showed similar performance, which is quite interesting point to consider. This means that data of patients, having larynx pathology, and normal subjects can be separated quite well, even when regarding all samples as support vectors, which is the case in LS-SVM, which has lower computational cost for training than usual SVM. A preliminary explanation of this could be that the distances, we investigated, are suitable for detecting disordered voice.

While using the RBF and KPA kernels as a variant of GMM supervector, it was found that it is better to concatenate GMM means of several recordings, rather than use a single GMM representation of all subject’s recordings. Since for the distance kernels only a single GMM from all recordings of the subject was considered, aggregation of classification results on the decision level should be researched further, while using a separate GMM for each recording of a single subject.

The sequence kernel (KPA) and the distance kernels (KL-MCS and KL-EMD) outperformed the popular Gaussian (RBF) kernel. However, the difference in accuracy is statistically significant only in the distance kernels case. The KL-MCS kernel, using GMMs with full covariance matrices, and the KL-EMD kernel, using GMMs with diagonal covariance matrices, provided the best overall results.

The difference in performance between cases of full and diagonal covariance matrices was not too significant to be alarmed. Possible explanation why using GMMs with diagonal covariance matrices in the KL-EMD kernel gives better results could be that GMMs with full covariance matrices have significantly larger number of parameters and thus are more prone to over-fitting, i.e. non-robust parameter estimation due to the limited amount of training.
data. In Chapter 38.4 of Benesty et al. (2007) it is stated that ‘empirically we have observed that diagonal-matrix GMMs outperform full-matrix GMMs’. But this is not a general rule.

By automatically detecting the most appropriate number of Gaussian mixtures for each subject (or recording) and, therefore, reducing the model order (i.e. variable size GMMs instead of fixed), one could probably achieve a better accuracy and significantly lower the computational cost.

Appendix A. Supplementary data


References


