Correlation features and a linear transform specific reproducing kernel

Andreas Beschorner, Dietrich Klakow

Spoken Language Systems, Saarland University, D-66123 Saarbrücken, Germany
{andreas.beschorner,dietrich.klakow}@lsv.uni-saarland.de
http://www.lsv.uni-saarland.de

Abstract. In this paper we introduce two ideas for phoneme classification: First, we derive the necessary steps to integrate linear transform into the computation of reproducing kernels. This concept not restricted to phoneme classification and can be applied in a wider range of research subjects. Second, in the context of support vector machine (SVM) classification, correlation features based on MFCC-vectors are proposed as a substitute for the common first and second derivatives, and the theory of the first part is applied to the new features. Additionally, an SVM structure in the spirit of phoneme states is introduced. Relative classification improvements of 40.67% compared to stacked MFCC features of equal dimension encourage further research in this direction.

Key words: Correlation, Hilbert space, Reproducing kernel, Phoneme classification

1 Introduction

Established concepts like HMMs have long been in use for speech recognition and phoneme classification. In recent years systems have been influenced for example by generative models like GMMs [11], maximum a posteriori adaptive sequence estimation [5], discriminative methods [15] and along with the latter the theory of reproducing kernel Hilbert spaces (RKHS). In the context of reproducing kernels, sequence kernels [4] have been developed, capturing the non-static nature of speech or even modelling HMMs (see [13], pp. 430–436). Approaches like kernel combinations have been succesfully implemented in other fields of pattern recognition. [9], [10] and [7] solve the SVM-optimization considering convex and linear combinations of kernels on (heterogenuous) compound feature vectors.

In our work, we pursue a new approach for phoneme classification. First, looking at feature computation, we show how to embed linear mappings into reproducing kernels. Second, we redesign SVM classification strategies and adopt concepts of subphonemes/ phoneme states/ HMMs without touching the kernel or the need to solve a modified optimization problem.

The classification results encourage us to continue with research in this direction.

This paper is organized as follows. Subsequent to a brief review of the concepts of reproducing kernels and support vector machines, section 3 shows how linear mappings can be embedded into the evaluation of reproducing kernels. These theoretical derivations in part motivate section 4, in which we introduce MFCC-autocorrelation- and later cross-correlation features. Finally, we propose an SVM-based classification approach utilizing a representation in the spirit of phoneme states. To get comparable information, we refrain from using kernel combination for the new approach and compare classification results to experiments using stacked traditional MFCC-features (details in the respective section). The results of these experiments follow in section 5, and the paper closes with conclusions and perspectives in section 6.

2 Reproducing Kernels and SVMs

2.1 Reproducing kernels

The concept of reproducing kernels is based on the fact that any Hilbert space $\mathcal{H}$ on a set X of complex-valued, bounded functionals endowed with an inner product $\langle \cdot, \cdot \rangle$ admits a mapping $k : X \times X \to \mathbb{C}$ such that for all $\mathbf{z} \in X$:

(1) $k(\cdot, \mathbf{z}) \in \mathcal{H}$
(2) $\forall f \in \mathcal{H} : f(\mathbf{z}) = \langle f, k(\cdot, \mathbf{z}) \rangle$.

k is called a *reproducing kernel* and is unique within $\mathcal{H}$. It is easily verified ([2], [12]) that reproducing kernels defined as such are positive semidefinite (psd). Conversely, for every psd $k : X \times X \to \mathbb{C}$ there exists exactly one $\mathcal{H} \subset \mathbb{C}$ wherein k is a reproducing kernel. Property (2) is called the *reproducing property*, as the kernel reproduces the evaluation of the functional $f \in \mathcal{H}$ using the Hilbert space's inner product. Given such a k, the factorization lemma ([1]) implies the existence of a Hilbert space $\mathcal{H}$ and a function $\Phi : X \to \mathcal{H}$ such that $k(\mathbf{x}, \mathbf{z}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{z}) \rangle$.

A reproducing kernel k thus allows us to replace costly computations of a mapping Φ by an inner product in $\mathcal{H}$. Well known examples are the linear kernel $k_l(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T\mathbf{z}$, the polynomial kernel $k_{p^d}(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T\mathbf{z} + r)^d$ and the exponential kernel $k_e(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \left|\left|\mathbf{x} - \mathbf{z}\right|\right|^2)$. However, kernel functions are not generally restricted to numerical representations. Areas such as bioinformatics, data mining and part-of-speech-tagging in natural language processing, make frequent use of kernels defined on strings or on more complex data structures like trees.

2.2 Support Vector Machines (SVMs)

Given a linear separable two-class dataset, SVMs compute a hyperplane $\mathbf{w}$ seperating the two classes. The hyperplane is optimal in the sense that it has minimal margin amongst all hyperplanes separating the data, the margin being

the distance from $\mathbf{w}$ to any (training) sample.

Let $\mathcal{H}$ be an N-dimensional Hilbert space, M be the number of samples. Writing $\mathbf{x} = (x_1 \cdots x_N)$ for $\mathbf{x} \in X$ and $n = 1, \ldots, N$, let $\mathbf{w} \in \mathcal{H}$ and $\mathbf{x}_1, \ldots, \mathbf{x}_M$ be vectors in X. With $b \in \mathbb{R}$ being the bias or offset, $\{\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \,|\, \mathbf{x} \in \mathcal{H}\}$ is a subspace and hyperplane in $\mathcal{H}$ with normal vector $\mathbf{w}$. The dot product equals the length of the projection of either component onto the direction of the remaining one. Hence, the orientation of the hyperplane, $d(\mathbf{x}|\mathbf{w}) = \operatorname{sgn}(\langle \mathbf{x}, \mathbf{w} \rangle + b)$, is a useful decision criterion. For target labels $y_m \in \{\pm 1\}, m = 1, \ldots, M$, the products $y_m \cdot d(\mathbf{x}|\mathbf{w})$ classify samples $\mathbf{x}$ into either class 1 or -1. The optimization problem of finding he hyperplane is subject to one constraint for each training sample: $y_m \cdot d(\mathbf{x}|\mathbf{w}) \geq 1, m = 1, \ldots, M$. To achieve better generalization, it has been proposed ([3]) and, following them, [6]) to relax the constraints by introducing slack variables $\zeta_m \geq 1$, $m = 1, \ldots, M$, leading to soft margins. Using Langrangian multipliers α_m, $m = 1, \ldots, M$ to optimize under the constraints, the final dual form of the optimization problem,

$$\begin{aligned} &\underset{\boldsymbol{\alpha} \in \mathbb{R}^m}{\text{maximize}} \sum_{m=1}^{M} \alpha_m - \frac{1}{2} \sum_{m,l=1}^{M} y_m y_l \alpha_m \alpha_l \langle \mathbf{x}_m, \mathbf{x}_l \rangle \\ &\text{s.t.} \qquad \begin{cases} \langle \mathbf{y}, \boldsymbol{\alpha} \rangle = 0 \\ 0 \leq \boldsymbol{\alpha} \leq C \end{cases}, \end{aligned} \tag{1}$$

permits the substitution of the objective function's inner product by a kernel function k. The inequality $\boldsymbol{\alpha} \geq 0$ is to be understood elementwise, and the new decision function now is $d(\mathbf{x}|\mathbf{w}) = \sum_{m=1}^{M} \alpha_m y_m k(\mathbf{w}, \mathbf{x}_m) + b = 0$. As the dimension of the RKHS depends on the kernel used in (2.2), the data can be linearly separable in the RKHS even if this is not the case in the original space. For multiclass SVM cases, one-vs-one or one-vs-all strategies are commonly used. A thorough discussion is presented in [14] or [12].

3 Linear mappings and reproducing kernels

Let us consider a continuous, linear mapping $T : X \to Y$ between two Banach spaces X, Y. A basic result from functional analysis is that the space $X \oplus Y$ with a norm given by $||(x, y)|| = \sqrt{||x||^2 + ||y||^2}$, $x \in X, y \in Y$ is again a Banach space. The graph $G(T) = \{(x, Tx) \,|\, x \in X\}$ of T is a closed subspace of $X \oplus Y$, the norm consequently being $||x||_T = \sqrt{||x||^2 + ||Tx||^2} \geq ||x||$. In Hilbert spaces, $G(T)$ as a closed subspace is itself a Hilbert space, it's inner product defined on the concatenation of the components: Let $\mathcal{H}, \mathcal{H}_T$ be Hilbert spaces, $p, q \in \mathcal{H}$ and $T : \mathcal{H} \to \mathcal{H}_T$ with respective inner products $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{H}_T}$. Then $G(T) \subset \mathcal{H} \oplus \mathcal{H}_T$, and

$$\langle (p, Tp), (q, Tq) \rangle_{G(T)} = \langle p, q \rangle_{\mathcal{H}} + \langle Tp, Tq \rangle_{\mathcal{H}_T} . \tag{2}$$

In this conetxt, a well-known theorem from Riesz ensures that for every bounded, linear continuous operator $T : \mathcal{H} \to \mathcal{H}_T$ between a finite dimensional Hilbert space $\mathcal{H}$ and a Hilbert space $\mathcal{H}_T$ there exists exactly one adjoint operator $T^* : \mathcal{H}_T \to \mathcal{H}$ such that for all $p \in \mathcal{H}, q \in \mathcal{H}_T$ the equation $\langle Tp, q\rangle_{\mathcal{H}_T} = \langle p, T^*q\rangle_{\mathcal{H}}$ holds.

In our work we build on this theorem and, using the bilinearity of inner products in $\mathbb{R}$, recast equation (2) as follows:

$$\begin{aligned}\langle (p, Tp), (q, Tq)\rangle_{G(T)} &= \langle p, q\rangle_{\mathcal{H}} + \langle Tp, Tq\rangle_{\mathcal{H}_T} \\ &= \langle p, q\rangle_{\mathcal{H}} + \langle p, T^*Tq\rangle_{\mathcal{H}} \\ &= \langle p, q + T^*Tq\rangle_{\mathcal{H}} \\ &= \langle p, (I_{\mathcal{H}} + T^*T)q\rangle_{\mathcal{H}}\,, \end{aligned} \tag{3}$$

where $I_{\mathcal{H}}$ is the neutral element (that is, the identity matrix) of the endomorphisms of $\mathcal{H}$ and the last inner product is defined on $\mathcal{H} \times \mathcal{H}$. As $z^*(T^*T)z = (z^*T^*)(Tz) = (Tz)^*(Tz) \geq 0$, T^*T is psd and pd whenever the trace of T^*T does not equal zero. In this case, $(I_{\mathcal{H}} + T^*T)$ will be pd and a new reproducing kernel integrating T is given by $k_{T^*T}(p, q) = p^*(I_{\mathcal{H}} + T^*T)q$.
Most important, the transform keeps vectors in the same space, allowing usual kernel combination techniques. In our experiments, we apply kernel composition by inserting the new kernel into an exponential one.

The effect of T on the kernel can be controlled further by scaling the inner products $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ and $\langle\cdot,\cdot\rangle_{\mathcal{H}_T}$ by numbers $w_{\mathcal{H}}$ and $w_{\mathcal{H}_T}$ respectively. Due to the linearity of inner products, this leads to

$$w_{\mathcal{H}} \langle p, q\rangle + w_{\mathcal{H}_T} \langle Tp, Tq\rangle = \langle p, (w_{\mathcal{H}} I_{\mathcal{H}} + w_{\mathcal{H}_T} T^*T)q\rangle_{\mathcal{H}}\,,$$

Note that $w_{\mathcal{H}}, w_{\mathcal{H}_T} > 0$ will ensure positive definiteness. For T unitary, equation (3) reduces to scaling q as

$$\langle (p, Tp), (q, Tq)\rangle_{G(T)} = \langle p, ((w_{\mathcal{H}} + w_{\mathcal{H}_T})I_{\mathcal{H}})\, q\rangle_{\mathcal{H}}\,. \tag{4}$$

4 MFCC-Correlation Features

In this section we introduce new features consisting of usual MFCC-vectors plus its correlation with adjacent vectors, the latter replacing the widely used Δ and $\Delta\Delta$. In our setting, both parts of such vectors are convex combined via reproducing kernels during training and classification. While different components of MFCC feature vectors (and thus the different frequency subbands they represented) are decorrelated via a discrete cosine transform in the process of their computation, correlation remains within sequences of the same component. To keep the number of features reasonable, we only consider immediately adjacent neighbourhoods and MFCC features of length L.

Formalizing this, let $m^l_{n-1}, m^l_n, m^l_{n+1}, m^l_{n+2}$, $n \in \mathbb{N}$ be a sequence of adjacent MFCC vectors, where $l = 1, \ldots, L$ references a component of the vectors and the subindex n the speech frame the MFCC features were computed from. Using $\times$ to indicate cross correlation and forming two vectors, each of length three, of the same components of adjacent vectors, we get L cross correlation vectors $\tilde{m}_l$

$$\tilde{m}_l = \left(m^l_{n-1}, m^l_n, m^l_{n+1}\right) \times \left(m^l_n, m^l_{n+1}, m^l_{n+2}\right).$$

Normalising and stacking the $\tilde{m}_l$ finalizes the computation of the autocorrelation feature vector $\left(\tilde{m}^1, \ldots, \tilde{m}^L\right)$.

4.1 Linearization and a phoneme state like approach

Given a fixed vector x of finite length, correlation with any finite vector y is a transform linear in y. However, this is not true for the autocorrelation we compute in section 3. For this reason, we modify the process as follows. We simulate a phoneme state representation by splitting (training) samples of length s into start- and endsection (S and E respectively) of length $s \div 3$ and a middle section (M) of length $(s \div 3) + (s \mod 3)$. For each specific phoenem/class, we group those subfeatures, compute their averages, and denote these centers by x_S, x_M and x_E.
In a first approach we allow seven *SME*-based states: *SSS, SSM, SMM, MMM, MME, MEE, EEE*. They represent the position in a phoneme, and training sets are built based on this segmentation . Using the new vectors, we switch from auto- to crosscorrelation, applying the theory following immediately. Figure 1 illustrates, which 3-vector sequences of a phoneme sample contribute to which training set.

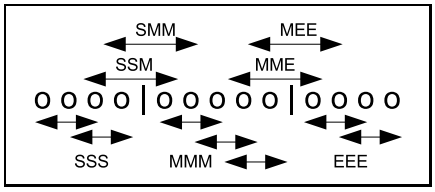

Fig. 1. Example of a phoneme sample comprised of 13 MFCC-vectors/ frames and its split into *SME*-based parts. Two vectorsequences are used for the classes *SSS* and *EEE*, three for *MMM* and one each for *SSM, SMM, MME* and *MEE*.

In comparison to autocorrelation, the computation now depends on the phoneme state. Considering a vector coordinate $1 \leq l \leq L$, the linear mapping is computed via $\left(x^l_{qs}, x^l_{qm}, x^l_{qe}\right) \times \left(m^l_n, m^l_{n+1}, m^l_{n+2}\right)$, where $qs, qm, qe \in \{S, M, E\}$.

In matrix form (omitting the index l for clarirty), the linear transform equals

$$T = \begin{pmatrix} 0 & 0 & x_{qs} \\ 0 & x_{qs} & x_{qm} \\ x_{qs} & x_{qm} & x_{qe} \\ x_{qm} & x_{qe} & 0 \\ x_{qe} & 0 & 0 \end{pmatrix}$$

and along with this

$$T^*T = \begin{pmatrix} x_{qs}^2 + x_{qm}^2 + x_{qe}^2 & x_{qs}x_{qm} + x_{qm}x_{qe} & x_{qs}x_{qe} \\ x_{qs}x_{qm} + x_{qm}x_{qe} & x_{qs}^2 + x_{qm}^2 + x_{qe}^2 & x_{qs}x_{qm} + x_{qm}x_{qe} \\ x_{qs}x_{qe} & x_{qs}x_{qm} + x_{qm}x_{qe} & x_{qs}^2 + x_{qm}^2 + x_{qe}^2 \end{pmatrix}.$$

If, for instance, the state in question is SSM, x_{qs} and x_{qm} take values from the cluster center vector x_s, whereas x_{qe} is set to the respective coordinate of x_m. Following the derivation of the previous section, T^*T will be pd and hence k_{T^*T} a reproducing kernel whenever $x_{qs}^2 + x_{qm}^2 + x_{qe}^2 \neq 0$.

5 Experimental Results

We present results from two sets of multiclass classification results. Section 5.1 refers to the autocorrelation features introduced in section 4 and part 5.2 depicts the experiments of the cross correlation setup described in section 4.1. The features were extracted via HTK3.3 extracted with a framesize of 25ms and an overlap of 10ms. Training and test were performed on the eleven most frequent phonemes *aa, ae, ay, eh, ey, ih, ix, iy, n, s, z* of the TIMIT dataset using a modified version of *svmlight* ([8]). If not mentioned otherwise, *svmlight*-parameters remained unchanged. Also, parameters for SVMs trained on the new vectors were not optimized but chosen due to results from partially rough grid tests. Evaluating on finer grids and, in the case of kernel combination, solving the convex kernel combination SVM optimization problem will very likely improve results further.

5.1 Autocorrelation Features

For the SVM-classification baseline we use standard MFCC features consisting of 13 values plus Δ and $\Delta\Delta$, and a single exponential kernel. γ is set to 0.001, a quasi-optimal value determined by a rough grid search. This is compared to SVM-classification using the autocorrelation features. Two exponential kernels on the two parts forming the vector – the 13 basic values and the 26 autocorrelation values – are convex combined with unchanged $\gamma = 0.001$. Using a convex weighting $wk_e^{mfcc} + (1-w)k_e^{corr}$ and a 1-vs-1 setup, we perform a rough first evaluation for $w = 0.05, 0.10, \ldots, 0.95$. Table 1 illustrates the results.

phoneme	*aa*	*ae*	*ay*	*eh*	*ey*	*ih*	*ix*	*iy*	*n*	*s*	*z*	**avg.**
MFCC39	38.7	73.4	63.9	64.2	65.8	94.1	25.2	22.9	44.8	12.4	88.4	**54.6**
MFCC13+corr	32.6	66.8	70.1	61.2	64.2	89.9	42.3	26.1	26.7	5.8	77.9	**51.2**

Table 1. Classification error rates results for two experiments: Features of size 39 (13 MFCC-values plus Δ and $\Delta\Delta$) using one RBF-kernel and features of size 39 (13 MFCC-values plus autocorrelation values) using a kernel combination. The latter produces a slight drop of the cumulative average classification error for all w. $w = 0.95$ gives the best results so far: A relative classification improvement of about 7.5%. For two classes, *ix* and *ay*, classification abates, while elsewhere it improves.

	sMfcc	SSS	SSM	SMM	MMM	MME	MEE	EEE	**SME-avg.**
aa	**69.26**	75.53	87.72	92.01	94.08	95.72	95.91	91.82	**90.40**
ae	**59.79**	94.27	83.54	83.71	92.66	79.65	79.96	88.76	**86.08**
ay	**52.22**	83.30	86.23	91.77	96.61	91.94	80.72	82.66	**87.60**
eh	**44.22**	62.54	88.44	91.44	92.23	90.95	87.68	65.58	**82.69**
ey	**56.39**	82.77	81.22	83.04	92.38	83.64	84.10	90.47	**85.37**
ih	**37.82**	79.72	77.50	71.74	75.85	73.40	83.07	90.41	**78.81**
ix	**47.63**	40.17	84.30	94.16	86.05	95.87	92.02	21.12	**73.38**
iy	**77.44**	96.21	96.66	96.72	97.68	96.40	96.03	91.32	**95.86**
n	**88.63**	95.71	97.96	98.23	97.79	98.35	97.36	88.57	**96.28**
s	**88.43**	98.59	96.35	94.89	96.15	91.88	92.36	99.37	**95.66**
z	**42.50**	77.45	87.39	79.38	48.61	66.60	63.04	21.01	**63.35**
avg.	**60.39**								**84.95**

Table 2. Recognition rates of *SME*-based classification compared to sMfcc features. Even phonemes like *ih* and *ix* that are hard to tell apart and often merged in experimental setups ([15], e.g.) are separated relatively well. The overall relative recognition gain is approximately 40.67%

5.2 Crosscorrelation Features and Phoneme Sate Simulation

As the new correlation features extend over three frames, comparison to single-frame MFCC-features is improper due to the difference in the amount of information. We thus consider 3-vector **s**equences of standard 13-dimensional **MFCC**-features (sMfcc) without Δ and $\Delta\Delta$, resulting in comparable feature vectors of equal dimension. To get a first impression of the quality of this new approach, we use a single exponential kernel. The γ-parameter is again selected due to a rough grid search and set to 0.0001 for the sMfcc-SVMs and to 0.00001 for the *SME*-SVMs. Table 2 illustrates the strong raise in the recognition rates.

6 Conclusion, Discussion and Perspectives

In this paper we have introduced correlation features computed from adjacent frames of MFCC-vectors of utterances and derived a kernel that integrates a linear mapping. Experiments using the new features-kernel combination show

great improvements in phoneme classification and encourage further research.

Clearly, not all phonemes are adequately represented by decomposition into all of the seven states or do not even deliver samples for the *SME*-trainingsets due to their size. Entries in table 2 like the EEE results for *ix* and *z* reflect this situation. Hence, individual setups are currently evaluated. Following those, a first step in classification will then be to perform intra-class evaluations and choose the one (or even two) classes with best recognition rates for further between-class classification. We are positive that this will again improve the results.
Finally, the definition of a graph also holds for operators, and equation (2) in section 3 can be recast in a similar way. In this context, the theory presented here becomes interesting for functions known to be reproducing kernels of for instance Sobolev- and Hardyspaces. For both operators and mappings, care must however be taken that $\langle Tf, Tg \rangle$ remains an inner products. Differentiation for instance annihilates its definitness.

References

1. J. Agler and K. McCarthy. *Pick Interpolation and Hilbert Functions Spaces*, volume 44 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.
2. N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.
3. K.P. Bennet and O.L. Magasarian. Robust linear programming discriminaion of two linearly inseparable sets. *Optimization Methods and Software*, 1:22–34, 1992.
4. W. M. Campbell. A sequence kernel and its application to speaker recognition. *Neural Information Processing Systems*, Vol. 14:1157 – 1163, 2001.
5. S. Chakrabartty and G. Cauwenberghs. Forward-decoding kernel-based phone sequence recognition.
6. C. Cortes and V. Vapnik. Support vector machines. *Machine Learning*, 20:273–297, 1995.
7. T. Hirokai et al. Simple but effective methods for combining kernels in computational biology. *RIVF*, pages 71 – 78, 2008.
8. T. Joachims. Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning*, 1999.
9. Gert R. G. Lanckriet et al. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning*, 5:27 – 72, 2004.
10. Gert R. G. Lanckriet et al. A statistical framework for genomic data fusion. *Bioinformatics*, 20:2626 – 2635, 2004.
11. E. Rodríguez et al. Speech/speaker recognition using a hmm/gmm hybrid model. *Lecture Notes in Computer Science*, 1206:227 – 234, 1997.
12. B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.
13. J. Shawe-Taylor and N. Chrstianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
14. A. Shigoe. *Support Vector Machines for Pattern Classification*. Advances in Pattern Recognition. Springer, 2005.
15. N.D. Smith and M. J. F. Gales. Using svms and discriminative models for speech recognition. *IEEE International conference on accoustic speech and signal processing*, 1:I– 77 – 80, 2002.