# FUZZY STRING KERNEL REPRESENTATIONS IN SPEECH PROCESSING

**Robert Kirchner**
Linguistics Dept., University of Alberta, Edmonton, AB, T6G2E1, kirchner@ualberta.ca

Many widely used approaches to pattern recognition/machine learning, including neural nets, k-nearest-neighbours classifiers, and support vector machines, have hitherto made little headway in speech processing, largely due to their inability to represent and compute over the variable-length sequential data of speech signals. A new technique, the string (subsequence) kernel, first applied in bioinformatics [1] and text classification [2], and extended to speech recognition by Goddard et al. [3], maps a variable-length input signal to a fixed-length feature array, by taking the inner product of n-gram subsequences. Similarity of signals can then be evaluated, by any of the above approaches, in the kernel space. In this presentation, two variations on Goddard's approach are considered and evaluated: a string kernel using fuzzy rather than absolute k-means clustering; and a kernel in which the feature counts are preserved as waveforms rather than scalars, to address the reverse mapping problem.

## 1. BASIC TECHNIQUE

Goddard vector quantizes a cepstral representation of the speech signal to a string of prototypes by k-means clustering. Assuming for illustrative purposes a mere three prototypes, all possible 2-grams are represented in a 3×3 matrix. The string {21313}, for example, contains the 2-grams {21} and {13}, and {31}. The corresponding cells of the matrix receive a value of 1 for each occurrence in the string ({13} occurs twice). Non-contiguous subsequences (e.g. {2…3}) are counted as well – allowing for detection of similarity notwithstanding interruption in the string – albeit with a value that exponentially decays with the distance between the elements. Multiple values within a cell are summed; and 0 is assigned to all other cells. The string thus maps to the kernel representation in Figure 1.

| 2nd \ st | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0.3679 *=exp(-1)* | 0 | 2.1353 *=1+1+exp(-2)* |
| 2 | 1 | 0 | 0.4177 *=exp(-1)+exp(-3)* |
| 3 | 1 | 0 | 0.3679 *=exp(-1)* |

**Figure 1.** 2-gram string kernel representation of the prototype string [21331].

In sum, this technique proceeds from the intuition that *two strings are similar to the extent that they contain the same subsequences*. Thus, the string kernel for {213} will have similar values to {21313}, even though the original strings are of different lengths, as it contains many of the same subsequences; while {22222} will have radically different values. This approach can be extended to n-grams simply by assigning values to an n-dimensional array rather than a matrix. Goddard reports that on a multi-voice (Spanish) digit recognition task, with 32 prototypes, an SVM classifier using this kernel outperforms a discrete HMM.

## 2. FUZZY K-MEANS CLUSTERING

A possible alternative to Goddard's vector quantization step would be to construct multiple string kernels for a given input, one for each frequency (or quefrency) channel. This approach proved unworkable, presumably due to its failure to detect patterns relating subsequences in one channel to simultaneous events in other channels. Vector quantization, by comparison, provides a useful 'gestalt' of each frame. On the other hand, vector quantization loses intra-prototype differences in the signal. Furthermore, frame A may be somewhat similar to frame B, but highly dissimilar to frame C; but under vector quantization, unless A and B happen to fall within the same prototype, the greater distance AC vs. AB is lost. It was hypothesized that recognition in the kernel space would improve, using fuzzy k-means clustering. In this fuzzy kernel approach, each frame receives fuzzy membership threshold scores for all the prototypes, and these strings of prototype scores are then mapped to 2-gram kernel space. That is, the kernel consists of multiple channels, one for each prototype, encoding counts of 2-grams such as 'membership > 0.8 precedes membership > 0.1 in prototype 6.' The resulting representation provides a gestalt of the frame; but because it encodes fuzzy membership of frames to multiple prototypes, this kernel is less lossy.

To test this hypothesis, recordings were made of 6 English speakers, 4 male and 2 female, saying 100-200 tokens each, in random order, of the English digits, for a total of 1000 tokens. In Matlab, the sound files were segmented into isolated digits, and converted into spectral and cepstral form with the RastaMat toolbox [4], an implementation of RASTA-PLP filtering [5]. Cepstral representations were initially used, as in [3]; but as this method achieved recognition rates below 30% on the training data, cepstra were abandoned in favour of spectral frames. Absolute and fuzzy k-means clustering (with 32 prototypes) and mapping to string kernel form were each implemented in Matlab [6]. 66% of the data were used for training, and 33% as test data. Similarity between kernels was measured as the negative exponential of their Euclidean distance, and a k-nearest-neighbour classifier was applied to these the similarity scores from these kernel representations.

Contrary to the hypothesis, Goddard's string kernel with absolute k-means clustering outperforms the fuzzy kernel (Table 1).

| | 2-gram | 3-gram |
|---|---|---|
| Goddard's kernel | 83 | 91 |
| Fuzzy kernel | 74 | 79 |

**Table 1.** Percent correct on knn classification (k=8) for test set

A possible explanation for this result lies in the ubiquity of the features on which the kernel is based, the fuzzy prototype membership threshold scores. In attempting to extract similarities or differences beyond prototype labels, the fuzzy kernel method, it seems, gets swamped with slight similarities, to the point that critical differences are obscured. This result is somewhat disturbing, as it suggests that the relatively poorly understood vector quantization process is not merely a convenient way of reducing the dimensionality of the data, but a crucial part of getting patterns to emerge from the data under this approach. Further research is required to determine whether some variant use of fuzzy clustering in string kernels, resulting in less ubiquitous features, might yield higher recognition scores.

## 3. REVERSE MAPPING WITH WAVE-FORM KERNELS

While Goddard's string kernel appears to work well for perceptual classification/recognition, to develop this approach into a general model of speech processing, i.e. including synthesis, it must be possible to reconstruct the original (prototype string) representation from the kernel. While this issue is perhaps not of immediate concern to engineers in the speech recognition industry, it is of concern to phonological theory, as it has been suggested in a growing body of research, e.g. [7-11], that a quasi-exemplar-based model of speech processing affords an elegant account of a range of lexical frequency effects in phonological and grammatical patterning, as well as an explicit learning algorithm whereby patterns of phonetic variation become entrenched as phonological constraints. Unfortunately, the Goddard kernel is not reversible (John Goddard, p.c.). A single occurrence of a contiguous subsequence cannot be distinguished from multiple occurrences of a non-contiguous subsequence, once the counts have been summed.

A technique for preserving the component counts notwithstanding summation is to treat them not as scalars, but as amplitudes of a complex waveform. Specifically, for each subsequence {AB} where A occurs in frame *f* in the vector quantized representation, a fixed-length sine wave of frequency *f* is created. The amplitude of this wave is the count of the (contiguous and non-contiguous) occurrences of this subsequence in the vector quantized representation. Now, if there are multiple occurrences of {AB}, with A occurring in several different frames, the corresponding waves can be summed to a single complex waveform for cell {AB}; the affiliation of A to its original timeframe(s) is retained in the component frequencies of the waveform. To map the string kernel back to the vector quantized representation, one merely has to identify these frequencies, by Fourier analysis, and to assign each prototype with a particular frequency to the corresponding frame. An approximation of the original spectrogram can then be obtained by replacing each prototype label with the spectral values of the corresponding prototype centre, as computed during the vector quantization process.

To test whether the representation of subsequence counts in waveforms rather than scalars reduces accuracy in recognition, waveform kernels for the training and test data were constructed, as described above. For reasons of computer memory limitations, only 2-gram kernels were evaluated. Results of knn classification, compared to those of Goddard's kernel, are not yet available as of the date of submission of this summary paper. However, preliminary results on the training data suggest that the waveform kernel representation achieves recognition rates at least as high as those of Goddard's kernel.

## References


[1] C. Watkins, Kernels from matching operations, CSD-TR-98-07, Royal Holloway, U London, 1999

[2] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, Text classification using string kernels, JMLR 2:419-444, 2002

[3] J. Goddard, A. Martinez, F. Martinez, H. Rufiner, A comparison of string kernels and discrete hidden Markov models on a Spanish digit recognition task, JASA 112:5, pt. 2, 2002

[4] Dan Ellis, RastaMat Toolbox, http://labrosa.ee.columbia.edu/matlab/rastamat.

[5] H. Hermansky, N. Morgan, RASTA processing of speech, IEEE Transactions Speech & Audio Proc, 2:4, 578-589, 1994

[6] Robert Kirchner, Exemplar Toolbox, http://www.ualberta.ca/~kirchner/.

[7] K. Johnson, Speech perception without speaker normalization, in Talker Variability, Academic Press, 1997.

[8] R. Kirchner, Preliminary thoughts on phonologization within an exemplar-based speech processing system, UCLAWPL 6, 1999.

[9] J. Bybee, Phonology and language use, Cambridge U. Press, 2001.

[10] J. Pierrehumbert, Word-specific phonetics, Laboratory Phonology VII, 101-140, Mouton de Gruyter, 2002.

[11] R. Kirchner, C. Fraser, Modelling phonologization within an exemplar-based lexicon, poster, 3rd Intl. Conf. Mental Lexicon, Banff, Oct 7, 2002.