

# Online feature selection for contextual time series data (Extended abstract)

Petteri Nurmi and Patrik Floréen

Helsinki Institute for Information Technology HIIT  
Basic Research Unit  
Department of Computer Science, P.O. Box 68  
University of Helsinki, FI-00014 Finland  
{firstname.lastname}@cs.helsinki.fi

**Abstract.** We propose a simple and efficient method for online feature selection from time series data. Our method is based on calculating characteristics of the different features and calculating similarity values for feature pairs using Gaussian kernels. Our motivation has been to design a method that can be used to select the most relevant context features for activity recognition. Namely, traditional feature selection methods have been designed for offline use and thus are not applicable in our setting. The efficiency of our method is evaluated using toy data and real context data, gathered using a 3D accelerometer.

## 1 Introduction

Dimensionality reduction plays an important role, e.g., in pattern recognition and exploratory data analysis [GE03]. Due to the generic nature of the task, there is a large number of different methods proposed in the literature. In general, dimensionality reduction methods can be divided into two main categories [Tor03]: *feature selection* techniques attempt to reduce dimensionality by discarding some of the original features, whereas *feature transform* methods attempt to map the original features into a lower dimensional subspace. The main problem with feature selection techniques is that they are unable to find features that *jointly* maximize a predefined criterion [KS96]. On the other hand, feature transformation methods tend to be slower and they often reduce the interpretability of the data.

In the context of time series data, the existing dimensionality reduction methods fall into two categories. Firstly, wave-form techniques apply a discrete valued transformation on the original series, after which the most important coefficients are selected, spanning a space of smaller dimension than the original one. Methods falling into this category include the *singular value decomposition* (SVD), the *discrete Fourier transform* (DFT) and the *discrete Wavelet transform* (DWT) [TK04]. Secondly, approximation techniques attempt to find a discretized representation for the original series using piecewise constant functions. These techniques include the *piecewise linear approximation* (PLA) [MYAU01], the *piecewise aggregate approximation* (PAA) [KCPM01] and *(k,h)-segmentation* [GM03].

Our focus area is activity recognition in adaptive context-aware systems. In many contemporary systems, e.g. [DA00, RCG<sup>+</sup>03], a distributed approach for activity recognition is used. In those systems, separate feature extractors are often used to derive different kind of features from raw sensor measurements.

Unfortunately existing methods are inapplicable for adaptive context-aware systems, where the goal is to reduce the dimensionality of data in an online manner. In order to separate the application design from the framework design, maintaining the interpretability of the data is also important. We propose an algorithm that calculates a similarity matrix for different features and removes the redundant ones. The entries of the matrix are calculated using a combination of Kernel methods [STC04] and traditional time series analysis techniques [Ham94]. Our experiments show that our algorithm is capable of recognizing both periodic and trend-related similarities from the original time series.

The organization of the paper is as follows. Section 2 introduces the proposed method. In Sec. 3 we evaluate our algorithm in a practical setting and, finally, Sec. 4 concludes the paper and discusses future work.

## 2 Online feature selection for time series

In our setting, data is first gathered from a set of sensors. The sensors typically measure physical characteristics of the situation of the user such as accelerometer readings, location of the user (GPS or cell-ID), or the outside temperature. However, the potential amount of data that can be gathered from a single user is enormous and thus, in order to reduce communications and computational burden, the amount of transmitted data needs to be reduced.

The first level of reduction is achieved by calculating features that aggregate the signals over time. For instance, for an accelerometer with a sampling rate of 100 Hz, the features can be calculated using the 100 data points corresponding to the measurements gathered over one second. Typical features include the mean, variance, autocorrelation and absolute magnitude, which are calculated from these 100 data points. Each feature is derived using a separate software component.

The focus of this paper is on the problem of selecting which features are the most relevant ones and detaching automatically the redundant software components. The framework has been designed to be run, at least potentially, on a handheld device, and thus the computational and storage capacities of the algorithm need to be minimized.

Let  $\mathbf{x}(t) = \{x_1(t), \dots, x_n(t)\}$  be the vector of features available at time  $t$ , i.e. each component  $x_i$  represents a feature calculated by a different software component. The proposed algorithm works as follows: at each time step  $t$  a similarity value  $d_{i,j}(t)$  is calculated for each pair of features  $(x_i(t), x_j(t))$ . The similarities are summed over time and scaled to the interval  $[0, 1]$ , after which the resulting values can be used to remove the redundant features. In the following we first describe how the similarities  $d_{i,j}(t)$  are calculated and after that we discuss the other parts of the algorithm.

In order to calculate the similarities  $d_{i,j}(t)$  at each instance of time  $t$ , we derive a set of  $k$  characteristics  $\{f_k^i(t)\}$  for every component  $x_i$  of the original feature vector  $\mathbf{x}$ . These characteristics are motivated by typical properties of time series, namely growth and periodicity over time; features with similar growth (or decay) or periodicity give no additional information for the task of activity recognition.

Currently we use four characteristics, denoted  $f_1^i(t)$ ,  $f_2^i(t)$ ,  $f_3^i(t)$  and  $f_4^i(t)$ , for calculating the similarities. The first of these captures time series moving together by looking at the absolute magnitude of growth/decay in each step:

$$f_1^i(t) = |x_i(t) - x_i(t-1)|. \quad (1)$$

As our second characteristic we use the scaled difference from mean to capture variable scaling and growth related similarities. We maintain a running estimate of the mean value  $\hat{x}_i(t)$  of a time series and define:

$$f_2^i(t) = \frac{x_i(t) - \hat{x}_i(t)}{\hat{x}_i(t)}. \quad (2)$$

The last two characteristics are designed to capture similarities in the periodicity of the variables. A relatively straightforward way to do this is to first scale the values to the interval  $[-1, 1]$  (done online by maintaining an estimate of the maximum absolute value  $\max_{|x(t)|}$ ) and then use the arcsin and arccos functions. However, in order to capture the change of phase angles we take the difference of the arcsin and arccos functions between consecutive time steps. Thus we define our features as:

$$f_3^i(t) = \arcsin\left(\frac{x_i(t)}{\max_{|x_i(t)|}}\right) - \arcsin\left(\frac{x_i(t-1)}{\max_{|x_i(t)|}}\right), \quad (3)$$

$$f_4^i(t) = \arccos\left(\frac{x_i(t)}{\max_{|x_i(t)|}}\right) - \arccos\left(\frac{x_i(t-1)}{\max_{|x_i(t)|}}\right). \quad (4)$$

The features are transformed into similarity values using the Gaussian Kernel, defined in Eq. 5 below. We then combine the similarity values for our four characteristics to obtain at each time instance  $t$  a joint Kernel matrix  $K_{ij}(t) = \sum_{l=1}^4 k(f_l^i(t), f_l^j(t))$ , where

$$k(u, v) = \exp\left(-\frac{\|u - v\|^2}{2\delta^2}\right) \quad (5)$$

and  $\delta$  is the width of the Kernel. We used in all experiments the value  $\delta = 0.10$ , but by changing the parameter values, different characteristics can be weighted more or less. The overall similarity value at time  $t$ ,  $d_{i,j}(t)$ , is the entry  $i, j$  of the Kernel matrix  $K_{ij}$  divided by the number of time steps elapsed, i.e.

$$d_{i,j} = \frac{1}{t} \sum_{k=1}^t d_{i,j}(k) = \frac{1}{t} \sum_{k=1}^t K_{ij}(k). \quad (6)$$

$\sin(x)$	1.000	0.772	<b>0.998</b>	0.776	0.752	<b>0.984</b>
$\cos(x)$	0.772	1.000	0.775	<b>0.997</b>	0.750	0.768
$\sin(x) + 2.0$	<b>0.998</b>	0.775	1.000	0.779	0.756	<b>0.989</b>
$\cos(x) + 5.0$	0.776	<b>0.997</b>	0.779	1.000	0.755	0.773
$x$	0.752	0.750	0.756	0.755	1.000	0.750
$\sin(x + \pi)$	<b>0.980</b>	0.767	<b>0.989</b>	0.773	0.750	1.000

**Table 1.** Results for the toy data set.

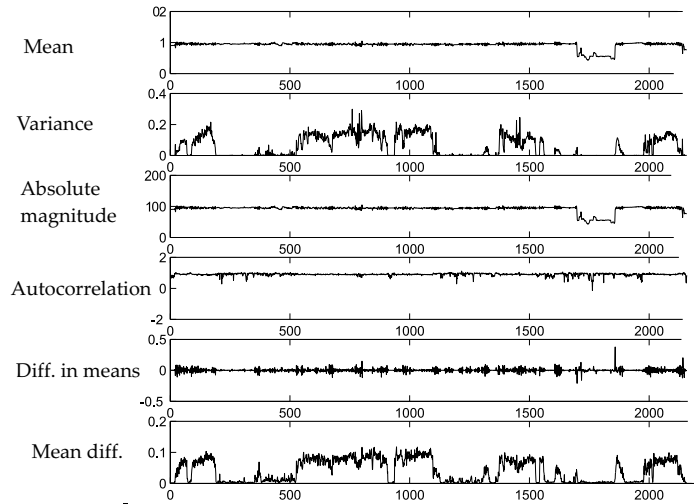
Now, the filtering of relevant features is relatively straightforward. Namely, the values in the symmetric matrix  $K$  represent pairwise similarities between the different features. We can now select the highest similarity (in the upper or lower diagonal matrix, ignoring diagonals) and remove either of the two corresponding features.

### 3 Experiments

The effectiveness of the algorithm was tested using two data sets. The first set was toy data, which was generated by first taking 1250 equally spaced points from the interval  $[0, 8\pi]$  and then six features were generated from these points by applying the functions  $\sin(x)$ ,  $\cos(x)$ ,  $\sin(x) + 2.0$ ,  $\cos(x) + 5.0$ ,  $x$ , and  $\sin(x + \pi)$ . The resulting similarity values given by Eq. 6 are given in Table 1. The algorithm correctly identifies (the values in bold in the table) the mutually translated features: 1,3 and 6; and 2 and 5.

The second data set consisted of 3-dimensional accelerometer signals gathered from a single user. The sampling rate of the accelerometers was 100 Hz and we aggregated the data over one second. The features used were *mean*, *variance*, *absolute magnitude*, *autocorrelation*, *difference in means* and *mean difference*. To be more specific, we took as one one data point for the feature vector *mean* the mean value  $\bar{s} = \sum_{i=1}^{100} s_i / 100$  of the 100 measures  $s_1, \dots, s_{100}$  corresponding to the measurements of one second. The in total about 200 000 signal measurements are thus reduced to about 2000 data points in the feature vector. Correspondingly, the *variance* is  $\sum_{i=1}^{100} (s_i - \bar{s})^2 / 99$ , the *absolute magnitude*  $\sum_{i=1}^{100} |s_i|$ , the *autocorrelation*  $\sum_{i=2}^{100} (s_i - \bar{s})(s_{i-1} - \bar{s}) / \sum_{i=1}^{100} (s_i - \bar{s})^2$  and the *mean difference*  $\sum_{i=2}^{100} |s_i - s_{i-1}| / 99$ . The *difference in means* is the difference between consecutive feature points of the first feature. Note that this all results in 18 feature vectors, six for each of the three dimensions. The data for the second experiment is illustrated in Fig. 1. In the figure, only the most relevant dimension, from perspective of detecting similarities, is shown for each feature.

The features to be detached were then detected as follows. The similarity values were checked every 30 time steps. If there were any similarity values above 0.9, one of the two features related to the highest similarity value was (randomly) selected and the similarity values reset to zero. The feature selected was removed in all three dimensions, i.e., if the mean was selected, all three feature vectors corresponding to the means in the three dimensions were removed.



**Fig. 1.** Plot of features derived from the accelerometer readings.

The experiment was conducted in a purely online fashion. The first similar pair identified by the algorithm was autocorrelation and mean, of which autocorrelation was eliminated. The difference in means was the second feature to be removed, the variance the third and the absolute magnitude the fourth. We then ran the algorithm several times and the elimination always resulted in two features, of which one is either the mean difference or variance and the other is one of the others. Even with the crude detachment procedure we used, the remaining two features, i.e. six feature vectors as there are three dimensions, made it possible in all cases to classify the known activities with over 90 % accuracy. This means that we would have reduced communication costs by two thirds in a mobile setting, which is a significant saving.

## 4 Conclusions and Future Work

In this paper we presented a novel technique for calculating similarities of time series and used it to filter features in context-aware systems. The method combines traditional time series techniques and Kernel methods and can be used in an online fashion. In addition, we illustrated that the algorithm can extract both periodic and trend-related similarities. Currently we use no non-linear similarity characteristics, but this is straightforward to implement. However, it seems that in practical applications (even with non-linear data), the current approach seems to work adequately.

## Acknowledgements

This work was supported in part by the IST Programme of the European Community, under the PASCAL network of excellence, IST-2002-506778. The publication only reflects the authors' views. The authors wish to thank Adrian Flanagan and Jukka Kohonen for performing the data recordings.

## References

- [DA00] Anind K. Dey and Gregory D. Abowd. The context toolkit: Aiding the development of context-aware applications. In *Proceedings of the Workshop on Software Engineering for Wearable and Pervasive Computing*, June 2000.
- [GE03] Isabelle Guyon and André Elisëff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157 – 1182, 2003.
- [GM03] Aristides Gionis and Heikki Mannila. Finding recurrent sources in sequences. In *Proceedings of the 7th International Conference on Research in Computational Molecular Biology*, pages 123 – 130. ACM Press, 2003.
- [Ham94] James D. Hamilton. *Time Series Analysis*. Princeton University Press, 1994.
- [KCPM01] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and Information Systems*, 3(3):263 – 286, 2001.
- [KS96] Daphne Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of the 13th International Conference on Machine Learning (ICML)*, pages 284 – 292, 1996.
- [MYAU01] Yuu Morinaka, Masatoshi Yoshikawa, Toshiyuki Amagasa, and Shunsuke Uemura. The L-index: An indexing structure for efficient subsequence matching in time sequence databases. In *Proceedings of the Fifth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2001)*, pages 51 – 60, 2001.
- [RCG<sup>+</sup>03] Diego Rios, Patricia Dockhorn Costa, Giancarlo Guizzardi, Luis Ferreira Pires, José Goncalves Pereira Filho, and Marten van Sinderen. Using ontologies for modeling context-aware services platforms. In *Proceedings of the Workshop on Ontologies to Complement Software Architectures (OOP-SLA)*, 2003.
- [STC04] John Shawe-Taylor and Nello Christiani. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [TK04] Sergios Theodoridis and Kostas Koutroumbas. *Pattern Recognition*. Academic Press, 2nd edition, 2004.
- [Tor03] Kari Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research*, 3:1415 – 1438, 2003.