An Alignment Approach for Context Prediction Tasks in UbiComp Environments

The authors detail the alignment prediction approach—a time-series-estimation technique applicable to both numeric and nonnumeric data—and compare it to four other prediction approaches to determine context prediction accuracy in ubiquitous computing environments.

Stephan Sigg
Technische Universität Braunschweig

Sandra Haseloff
Alexander von Humboldt Foundation

Klaus David
University of Kassel

In the field of pervasive computing, researchers use context prediction to infer future context information—classified and possibly aggregated features of sensor readings that describe an entity's situation.[1] Context prediction algorithms use the set of observed past and present contexts as input data. The possibilities of context-aware applications are then extended due to this widened situation awareness. To be feasible, context prediction requires that a distinguishable process underlies the observed context sequence. Application domains range from improved sensor network resource utilization to accident prevention to assistance in development processes. Prominent context-prediction techniques are Markov predictors,[2] SOM (self-organizing maps) prediction methods,[3,4] the state predictor method,[4,5] neural network approaches,[5,6] Bayesian networks,[5] prediction based on PCA (principal component analysis),[7] ARMA (autoregressive moving average) predictors,[4] and Kalman filter methods.[8]

Naturally, these algorithms require high prediction accuracy paired with long prediction horizons. To be suitable for ubiquitous environments, processing and memory requirements must be low and should have a reasonable error tolerance for input data. Also, algorithms should be able to handle multiple data types because context can be represented both numerically and nonnumerically (see Figure 1). Finally, a broader scope of context information is preferable because it usually provides a better situation description. However, a context-aware system is then likely required to handle multidimensional and multitype context sequences.

We compare the alignment approach to four other techniques to determine its suitability in various context prediction tasks in ubiquitous computing environments.

The Alignment Prediction Approach

Alignment prediction relies on typical context patterns. This method can handle multidimensional and multitype context sequences and has reasonable memory requirements and a scalable error tolerance. It uses alignment techniques that computational biologists have applied to find approximately matching patterns between RNA or DNA sequences. It can also abstract from process noise in the input sequence. This is possible because the similarity between observed and typical sequences is computed by a metric that rewards smaller deviations in context time series elements with smaller penalty costs.

Published by the IEEE CS ■ 1536-1268/10/$26.00 © 2010 IEEE

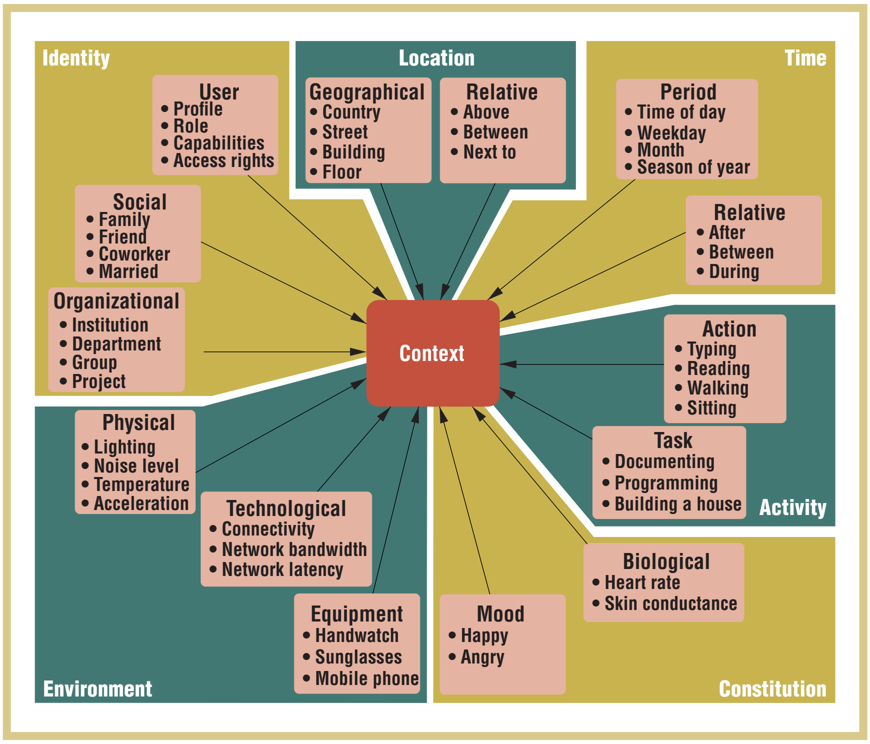


Figure 1. Several aspects of context. Because context can be represented both numerically and nonnumerically, context prediction algorithms should be able to handle multiple data types.

In alignment prediction techniques, most similar subsequences are computed between two patterns,[9,10] estimating the similarity between symbols σ of an alphabet $\sum$ using a metric $\delta : \sum \times \sum \rightarrow \mathbb{R}$. We can compute the optimal solution iteratively for increasing subsequence lengths.[11] For two patterns $p_1,\ldots,p_n$ and $q_1,\ldots,q_m$, we calculate the alignment with maximum weight using integer programming:

$$
\begin{aligned}
&w_{p_1 \ldots p_i, q_1 \ldots q_j} = \\
&\max(w_{p_i \ldots p_{i-1}, q_1 \ldots q_{j-1}} + \delta(p_i, q_j), \\
&\qquad w_{p_1 \ldots p_{i-1}, q_1 \ldots q_j} + \delta(p_i, -), \\
&\qquad w_{p_1 \ldots p_i, q_1 \ldots q_{j-1}} + \delta(-, q_j), \\
&\qquad\qquad 0).
\end{aligned}
$$

In this formula, the gap symbol "–" represents a pattern's missing symbol. When noise is induced by additional symbols in one sequence, the insertion of this gap symbol at the respective position of the other enables the comparison of both sequences regardless of noise or additional context values. The "0" is necessary to find an alignment between subsequences instead of complete sequences.

Context prediction using alignment prediction methods is a three-step process. First, we identify typical context patterns as either exact or approximate repeats in the observed context sequence. (See *Computational Molecular Biology—An Algorithmic Approach* for a summary of suitable algorithms for this task.[10])

Next, we compute semiglobal alignments between a suffix of the observed sequence and each typical sequence to identify a typical pattern. We apply the Needleman-Wunsch algorithm to calculate all semiglobal alignments between two context time series.[11] Alternatively, we can calculate semiglobal alignments using fast heuristics, for example, the Blast or Fasta algorithm.[9]

The final step is prediction. Up to this point, we've computed subsequences of typical patterns that are most similar to a suffix of the observed sequence. Because the compared sequences are considered typical, the continuation of the most similar typical subsequence is the prediction. Figure 2 illustrates a prediction computation. The matrix is filled from top to bottom and from left to right using the weight calculation detailed above. The alignment matrix's first row and column are initialized with 0 so that the algorithm ignores mismatches at the start of both sequences. The predicted pattern is a suffix of the typical sequence that starts after the index of the subsequence with the best alignment rating (maximum weight in row n).

The alignment method is particularly well suited to predict (approximately) reoccurring typical patterns. The prediction horizon's length is bounded by the typical patterns' length. The approach is robust against measurement inaccuracies, as optimal alignments might contain gaps and mismatches. The similarity metric δ controls the degree to which the algorithm tolerates measurement errors.

We can compute alignments in $O(k^2)$ steps, where k represents the maximum length of all context sequences considered. The predicted sequences' calculation requires

$$O\left(\sum_{i=1}^{k} i\right) = O(k^2)$$

operations in the worst case (that is, when all sequences are of identical maximum length k). With ρ typical patterns, the overall complexity is then $O((k^2 + k^2)\,\rho) = O(k^2\rho)$, because the observed sequence is aligned with every typical sequence. Using fast heuristics, the runtime decreases to $O(k\rho)$ with reasonable prediction accuracy.[12–14]

Note that the runtime doesn't depend on the number of distinct contexts but only on the length and count of typical sequences. Therefore, we propose the use of alignment prediction techniques for medium- to large-scale ubiquitous computing scenarios with typical patterns. The input sequence can be

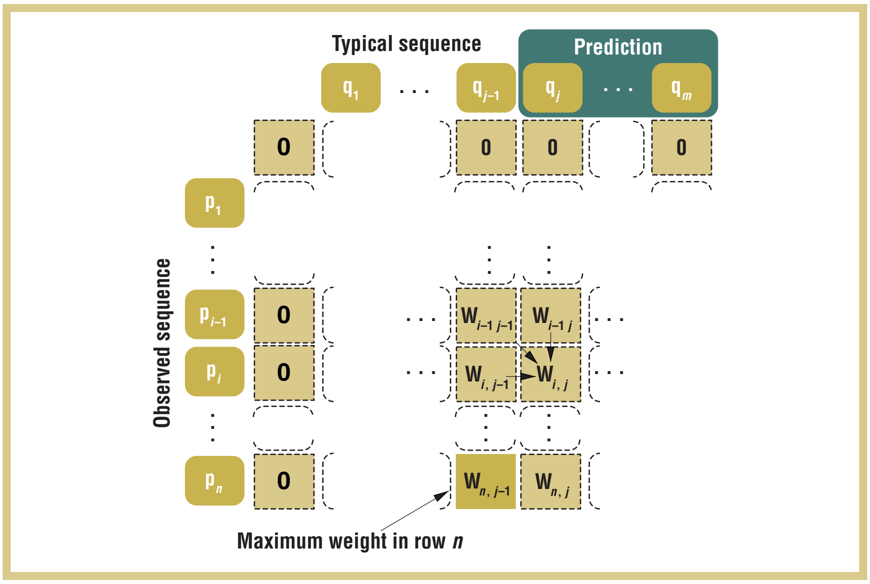


Figure 2. Calculation of a prediction based on an optimum alignment between input patterns. The first row and column of the alignment matrix are initialized with 0 so that the algorithm ignores mismatches at the start of both sequences. The predicted pattern is a suffix of the typical sequence that starts after the index of the subsequence with the best alignment rating (maximum weight in row n).

numeric or nonnumeric, or even multidimensional, mixed-type contexts.

Context Prediction Algorithms

Here, we detail the four prediction approaches we use in our simulation studies—Markov, ARMA, PCA, and ICA (independent component analysis). (See *Development of a Novel Context Prediction Algorithm and Analysis of Context Prediction Schemes* for information on prediction algorithms' strengths and weaknesses.[1])

Markov Processes

Markov processes constitute a major branch in the theory of stochastic processes. They're popular for their simplicity and applicability to a diverse set of problems.

In Markov context prediction domains, contexts are represented by states, and transition probabilities for each pair of consecutive observations are represented by Markov chain state transitions. In an order-k Markov chain, each state represents k consecutive contexts (or observations), and transitions between states correspond to observations of following contexts. Given a recently observed context, the Markov algorithm computes a probability distribution for an observed context sequence's next outcome. Iterating this process extends the prediction horizon.

Markov prediction techniques are optimal in the sense that they can always achieve the highest possible prediction accuracy for infinite binary random sequences.[15] The model can be applied to numerical and nonnumerical data alike. However, a prediction that reaches further into the future implicitly utilizes already predicted contexts, which might decrease the prediction accuracy.

The runtime depends on the probability graph's size. When C is the set of different observed contexts, a most probable next state is computed in time $O(|C|)$ in the worst case (that is, all possible $|C|$ state transition probabilities are nonzero for this state). The most probable n future contexts are computed in time $O(n \bullet |C|^2)$.

Prediction approaches with similar properties include hidden Markov models (HMMs), conditional random fields (CRFs), and dynamic Bayesian networks.[16–18]

ARMA

Despite developments in nonlinear methods, the most common stochastic models in time-series prediction are linear, such as the autoregressive moving average processes.[19] Because ARMA processes are already designed to approximate numeric time-series development, the only requirement for ARMA to be applicable to context prediction tasks is that context types are numerical.

ARMA processes achieved excellent results in context prediction tasks.[4,20] Because it combines autoregressive and moving average time-series-estimation models, this approach is well suited to predict trends and periodic patterns. It's also applicable in both one-dimensional and multidimensional datasets. The computational complexity is $O(k \log(k))$, where k is an observed time series' length.[19] No preprocessing or separate learning is required.

However, because ARMA prediction approaches are only applicable to sequences of numeric contexts, they can't be used in many problem domains. Although we can avoid this restriction by mapping nonnumeric contexts to numeric contexts (for example, using binary indicator feature vectors), this can create drawbacks such as dimension inflation or loss of neighborhood relations.

Principal Component Analysis

PCA is a technique to identify patterns in high-dimensional data. It's used in face recognition and image compression and can highlight data similarities and differences.[21] The approach reduces the number of dimensions by which data is represented without losing characteristic information.

Basically, PCA computes the eigenvectors and eigenvalues of the input data's covariance matrix. Eigenvalues indicate the significance of the data description's corresponding eigenvector. A vector representing new sampled data is then transformed to a new basis spanned by the most relevant eigenvectors—the principal components.

To predict context, we apply PCA to the input data's binary indicator feature vectors. Then, we divide the input data into various vectors that represent common behavior patterns (for example, 24 hours to represent a whole day). Resulting principal components serve as representatives describing the dataset.[7]

PCA initializes the hours to be predicted in a binary feature vector with 0. This suffix has identical distance to all complete binary feature vectors' suffixes of the same length. The vector's prefix represents its similarity to other feature vectors from the training data. The vector undergoes a basis transformation to the vector space spanned by the principal components and is associated with the most similar principal component. The corresponding binary feature vector's continuation serves as a prediction.

The PCA-based approach achieved high prediction accuracy on a dataset with three context classifications (Home, Work, Elsewhere).[7] The method's runtime depends on the number of distinct contexts $|C|$ in a scenario, because the binary feature vector's length increases with this value. When M behavior patterns are characterized by κ samples, the method's runtime is $O(M \cdot (\kappa \cdot |C|)^2)$ for nonnumeric context patterns and $O(M \cdot \kappa^2)$ in scenarios with numeric input patterns only (no transformation to binary indicator feature vectors is required).[22] Because κ is typically a multiple of the context history size, k, we can calculate the runtime for nonnumerical input as $O(M \cdot (k \cdot |C|)^2)$ and for numerical input as $O(M \cdot k^2)$. Therefore, this method is best suited when the number of distinct contexts $|C|$ is reasonable ($|C| << k$), especially in scenarios with nonnumeric input patterns.

For the PCA-prediction approach, a priori knowledge of common behavior patterns' length and occurrence time is required. When typical patterns don't reappear consistently at similar times, the prediction accuracy is reduced. Whereas many patterns in ubiquitous settings exist that are static in nature (for example, people sleep at night, then have breakfast, go to work, eat lunch, go back to work, and finally come back home), other patterns—although typical—might not follow such a strong scheme. Consider, for example, having phone calls or meetings. Typically, the time of day, duration, and activity flow differ for these situations. Because the PCA prediction requires common behavior patterns of a predefined length, these situations are hard to predict accurately with a PCA approach.

Independent Component Analysis

ICA is applied to audio processing, biomedical signal processing, image processing, and telecommunications.[23] This method transforms data linearly into components that are maximally independent of one another while, at the same time, describing the data's relevant properties. Similar to PCA, ICA identifies components that describe properties. However, whereas the PCA reduces the data dimension, the ICA might reduce, increase, or sustain the data dimension.

ICA achieves context prediction in a similar way as PCA. The main difference is that ICA uses independent components instead of principal components. With $|C|$ contexts and M behavior patterns characterized by $\kappa = O(k)$ samples, the computational complexity is $O(M \cdot (k \cdot |C|) \log(k \cdot |C|) + M^2 \cdot k \cdot |C|)$.[24]

Experimental Studies

We detail the results we obtained for ARMA, Markov, PCA-based, ICA-based, and alignment prediction approaches in four simulations. Two scenarios contain numerical data only and are characterized by trends and periodic patterns and by typical sequences, respectively. The other two studies use nonnumeric data characterized by frequent, typical sequences with few and numerous labels, respectively. For the numeric datasets, we compare the prediction algorithms' accuracy using the RMSE (root-mean-square error) metric. For a predicted time series of length n, it's defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - d_i)^2}{n}}.$$

In this formula, p_i denotes the predicted value at time i and d_i is the value that actually occurs. In the nonnumeric scenarios, we measure accuracy using the percentage of correct predictions.

Wind Power Prediction

Wind farms consist of a multitude of wind turbines whose performance fluctuates and depends heavily on wind power. Because the power supply system can't handle these fluctuating power curves, methods that predict wind power are necessary to schedule the power expulsion.

Our dataset contained wind power samples from a German wind farm. Samples were recorded hourly from February 2004 to April 2005. We used three-quarters of the data samples as training data and the remaining part for the simulation. We use alignment, ARMA, and Markov prediction approaches on this dataset.

The alignment approach used a context history of six hours with one sample per hour. Markov prediction was based on an order-six Markov model. The ARMA approach required more information about the observed context pattern to accurately extract trends and periodic patterns. Even with 100 hours of context history, the ARMA algorithm was inferior to both Markov and alignment prediction (see Figure 3). We therefore used the complete context history (at least 750 hours for one prediction) for the ARMA approach. Prediction horizons ranged from 1 to 20 hours.

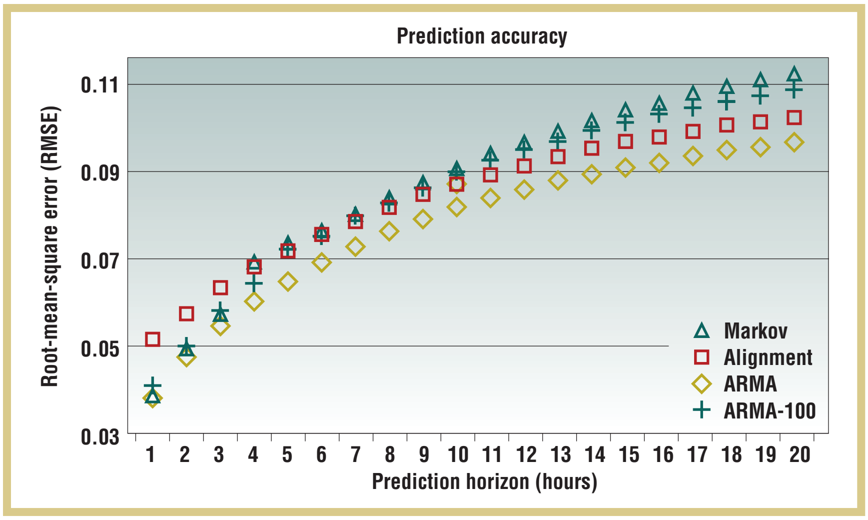


Figure 3. Prediction accuracies for the alignment, ARMA (auto regressive moving average), and Markov prediction algorithms. ARMA and Markov methods' accuracy are nearly identical for short prediction horizons, whereas the alignment prediction approach performs worse. With prediction horizons that exceed the context history size (more than six hours), alignment prediction outperforms Markov but doesn't reach the same accuracy as the unrestricted ARMA approach.

TABLE 1
Sampled GPS input fluctuations.

Distance (meters)	Average time to propagate distance (minutes)
5	6.866
10	8.9846
20	13.3859
30	17.175
50	24.9622
100	43.0246

We define the similarity metric as the difference between pairwise wind power samples. The alignment prediction approach used only one typical pattern—the wind power samples' complete training data. This accounts for the fact that the dataset length wasn't sufficient to cover periodic occurrences of patterns in distinct seasons. The underlying Markov chain for the Markov prediction approach was also based on the complete training set.

ARMA and Markov methods' accuracy are nearly identical for short prediction horizons, whereas the alignment prediction approach performs worse (see Figure 3). With prediction horizons that exceed the context history size (more than six hours), alignment prediction outperforms Markov but doesn't reach the same accuracy as the unrestricted ARMA approach. We conclude that the ARMA prediction approach is best suited in an environment characterized by trends and periodic patterns. However, with restricted context history information, the alignment prediction approach is better suited.

Location Prediction on Raw GPS Data

In the second study, we considered the prediction of a mobile user's location trajectory. We sampled an individual's GPS positions for 20 days. Our sampling hardware consisted of a mobile phone and a GPS receiver. The GPS receiver was connected to the phone via Bluetooth. We recorded GPS information every two minutes. When no GPS was available (for example, when the individual was indoors), we used the last available sample to approximate the position.

To obtain the optimal sampling frequency, we considered several sampling intervals. Table 1 depicts the average time observed for the user to propagate a fixed distance. We chose a sampling interval of 20 minutes to sample context changes but not the process noise.

We used the ARMA, Markov, and alignment prediction approaches and defined the alignment metric δ by the pairwise Euclidean distance between GPS measurements. Typical patterns in this study have been determined using alignment methods. When a pattern of sufficient length had an alignment rating below a defined threshold value, we considered this to be a new typical pattern. Also, the prediction algorithm frequently updated the Markov chain with new observed contexts. Otherwise, we used the same configuration as in the previous simulation for all three algorithms.

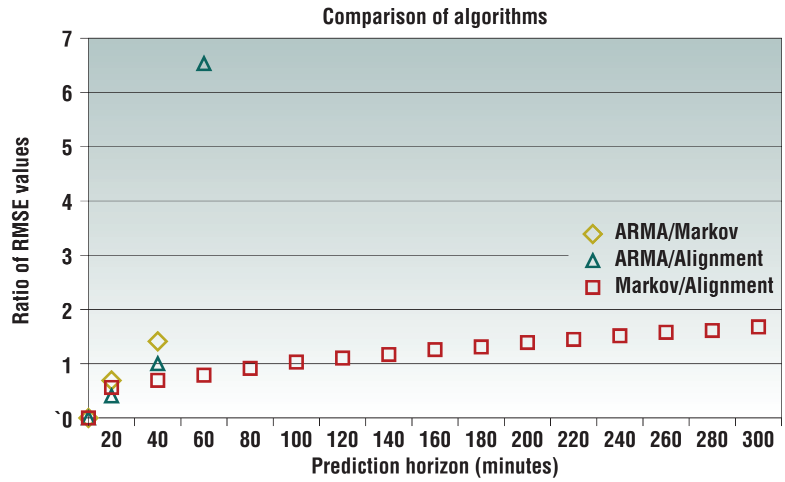
The ARMA approach isn't well suited for this dataset (see Figure 4). It achieves reasonably accurate predictions for short prediction horizons only. Whereas the first-order Markov prediction approach doesn't suffer from the data structure that much, prediction accuracy gradually decreases with an increasing prediction horizon. Because of its reliance on typical context sequences expected in the user's everyday behavior, the alignment prediction approach's accuracy remains quite stable for an increasing prediction horizon. Because the method computes a prediction on the basis of the similarity of observed and typical sequences, the prediction will either fail to match the actual sequence or will approximately match the sequence regardless of its length. The prediction accuracy is therefore nearly independent of the prediction horizon.

The Markov algorithm is more suitable for short prediction horizons of up to 80 minutes, whereas the alignment algorithm performs better for longer prediction horizons.

Location Prediction on Clustered GPS Data

We applied 36 labels—such as Market, University, and Home—to the raw GPS coordinates in the previous study. Every label was associated with a center position specified by GPS coordinates and a radius. Each GPS coordinate within this radius had a corresponding label; any GPS samples that didn't fall in any of these clusters were given the label Elsewhere. For the prediction process, we used a sampling frequency of 20 minutes and used the labeled locations' trajectory as input. We didn't provide GPS coordinates to the algorithms. Again, we measured the prediction approaches' accuracy using RMSE, where the labels' associated center locations are used for the similarity measurement.

We used a Markov prediction algorithm, an alignment prediction approach, and PCA- and ICA-based prediction methods. For PCA and ICA techniques, we achieved best results with typical behavior patterns that capture a whole day (72 samples or 24 hours). We used a default context history size of 16 hours. However, because the typical behavior patterns captured only 24 hours, we adapted the context history size by increasing the prediction horizon. The alignment prediction approach's context history size was set accordingly. We based the Markov prediction on an order-two Markov process.



Prediction horizon	5 min.	10 min.	15 min.
ARMA	2.36738	61527.8	2.8×10^7
Markov	0.08233	0.141960	0.19519
Alignment	0.1420	0.136650	0.13384

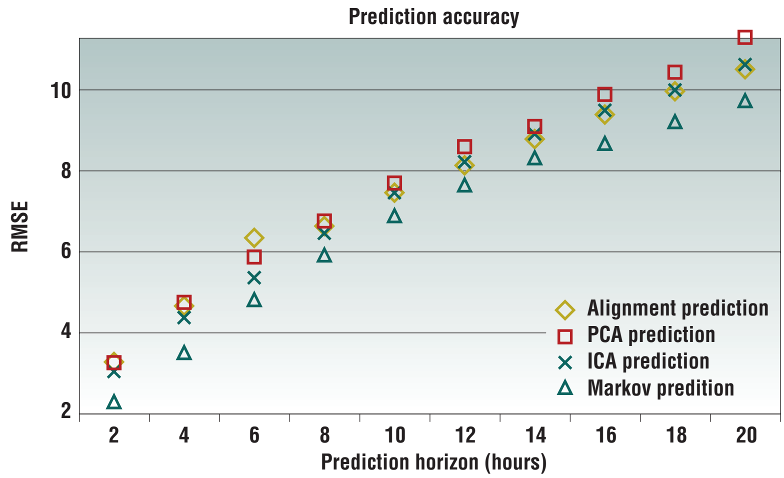
Figure 4. Markov, ARMA, and alignment prediction algorithms for the root-mean-square error (RMSE) metric with various sampling intervals and average error at various prediction horizons. The Markov algorithm is more suitable for short prediction horizons of up to 80 minutes, whereas the alignment algorithm performs better for longer prediction horizons.

In this scenario, with a reasonable number of nominal contexts linked together by their neighborhood relation, the Markov prediction approach has the best prediction accuracy (see Figure 5) because we can model the observed movement as a traversal between locations. This is exactly how the Markov prediction approach models the underlying state model. The ICA and alignment prediction approaches perform similarly in this scenario, whereas the PCA prediction technique's performance decreases with an increasing prediction horizon.

Prediction Based on Reality Mining Dataset

A PCA-based prediction approach achieves high performance on a dataset of cell-ID-based location information.[7] We applied the alignment, the PCA-based, and ICA-based prediction approaches on this reality mining dataset to predict mobile users' future locations.[25] The dataset holds 100 subjects' location, activity, and interaction information gathered from mobile phones over a nine-month time period. We chose an individual with a high number of data samples to predict location (GPS cell ID). In an online learning process, we used all previously observed contexts to train the prediction methods. Contexts are input at a frequency of 20 minutes. For all three approaches, the context history and the prediction horizon are 24 hours.

Because only the GSM towers' labels are given, we measure accuracy by the percentage of correctly predicted context labels, instead of the RMSE. We associate the Home, Work, and Elsewhere labels with distinct cell towers according to the amount of



	2 hours	10 hours	20 hours
Align	3.2834 (.309)	7.4192 (.654)	10.4923 (1.308)
PCA	3.2717 (.09)	7.6991 (.339)	11.2995 (1.013)
ICA	3.0801 (.08)	7.4344 (.2278)	10.6382 (.692)
Markov	2.0801 (.07)	6.8747 (.229)	9.7094 (.196)

Figure 5. Markov, alignment, PCA, and ICA algorithms' prediction RMSE (root-mean-square error) for clustered GPS data. RMSE and variance (in parentheses) are given for context prediction horizons of 2, 10, and 20 hours. The Markov approach has the best prediction accuracy. The ICA (independent component analysis) and alignment prediction approaches perform similarly in this scenario, whereas the PCA (principal component analysis) prediction technique's performance decreases with an increasing prediction horizon.

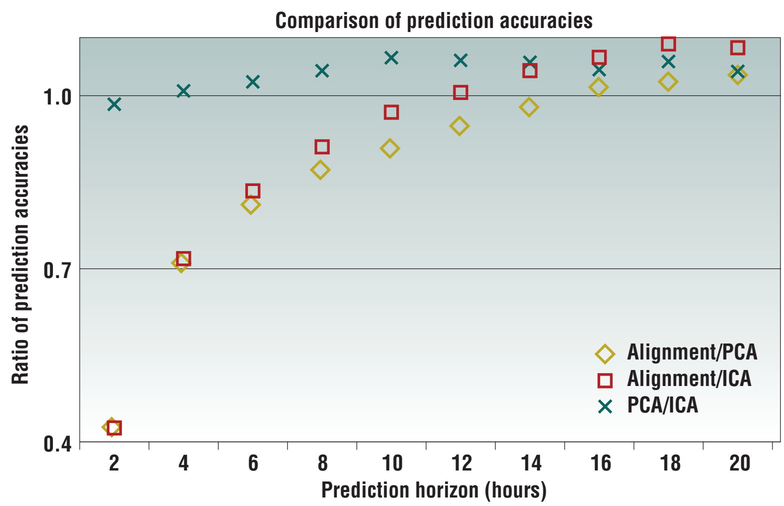


Figure 6. Prediction accuracy of the alignment, PCA, and ICA approaches for a subject in the reality mining dataset. For a short prediction horizon, the PCA- and ICA-based prediction approaches achieve better accuracy than the alignment prediction approach. However, as the prediction horizon increases, the alignment prediction approach performs better.

time spent at corresponding towers at meaningful times during a day. Figure 6 depicts the ratio of accuracy between two prediction techniques (that is, correct predictions of technique 1 to correct predictions of technique 2). For a short prediction horizon, the PCA- and ICA-based approaches achieve better accuracy than the alignment prediction approach. However, as the prediction horizon increases, the alignment prediction approach performs better.

We apply a variety of context prediction algorithms in different context prediction domains. A prediction algorithm's performance is always dependent on the input data's structure and type.

With our quantitative results, we expect that researchers will have a better understanding of prediction processes' impacts and make better algorithmic decisions in future studies and implementations. Whereas the ARMA prediction approach showed remarkable results on numeric-input time series incorporating trends and periodic patterns, the alignment prediction approach easily outperformed it in environments dominated by frequently reappearing typical patterns, especially for long prediction horizons. Markov prediction is especially well suited for short prediction horizons and dominated the other approaches in the scenario with many nominal contexts. Regarding the ICA- and PCA-based prediction approaches, the former achieved better results with many nominal contexts, whereas the latter performed better with fewer nominal contexts. However, both methods require a priori knowledge on behavior patterns' intermittency.

ACKNOWLEDGMENTS

The German Bundesministerium für Bildung und

Forschung (BMBF), in the framework of the Wireless Internet and mik21 projects, partially funded this project.

REFERENCES

1. S. Sigg, *Development of a Novel Context Prediction Algorithm and Analysis of Context Prediction Schemes*, University of Kassel Press, 2008.

2. K. Gopalratnam and D.J. Cook, "Active Lezi: An Incremental Parsing Algorithm for Sequential Prediction," *Int'l J. Artificial Intelligence Tools*, vol. 14, nos. 1–2, 2004, pp. 917–930.

3. G. Simon et al., "Time Series Forecasting: Obtaining Long Term Trends with Self-Organising Maps," *Pattern Recognition Letters*, vol. 26, no. 12, 2005, pp. 1795–1808.

4. R.M. Mayrhofer, "An Architecture for Context Prediction," doctoral dissertation, Institute for Pervasive Computing, Johannes Kepeler Univ. of Linz, 2004.

5. J. Petzold et al., "Prediction of Indoor Movements Using Bayesian Networks," *1st Int'l Workshop Location and Context Awareness* (LoCA 05), Springer, 2005, pp. 211–222.

6. M.C. Mozer, "Neural Net Architectures for Temporal Sequence Processing," *Predicting the Future Understanding the Past*, A.S. Weigend and N.A. Gershenfeld, eds., Addison-Wesley, 1994.

7. N. Eagle and A.S. Pentland, "Eigenbehaviors: Identifying Structure in Routine," *Behavioral Ecology and Sociobiology*, vol. 63, no. 7, 2009, pp. 1057–1066.

8. L. Capra and M. Musolesi, "Autonomic Trust Prediction for Pervasive Systems," *Proc. 20th Int'l Conf. Advanced Information Networking and Applications* (AINA 06), vol. 2, IEEE CS Press, 2006, pp. 481–488.

9. H.-J. Boeckenhauer and D. Bongartz, *Algorithmische Grundlagen der Bioinformatik,* Teubner, 2003 (in German).

10. P.A. Pevzner, *Computational Molecular Biology—An Algorithmic Approach*, MIT Press, 2000.

11. S.B. Needleman and C.D. Wunsch, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins," *J. Molecular Biology*, vol. 48, no. 3, 1970, pp. 443–453.

12. S.F. Altschul et al., "Basic Local Alignment Search Tool," *J. Molecular Biology*, vol. 215, no. 3, 1990, pp. 403–410.

13. S.F. Altschul et al., "Gapped Blast and Psi-Blast: A New Generation of Protein Database Search Programs," *Nucleic Acids Research*, vol. 25, no. 17, 1997, pp. 3389–3402.

14. W.R. Pearson, "Flexible Sequence Similarity Searching with the Fasta3 Program Package," *Methods in Molecular Biology*, vol. 132, no. 2, 2000, pp. 185–219.

15. M. Feder, N. Merhav, and M. Gutman, "Universal Prediction of Individual Sequences," *IEEE Trans. Information Theory*, vol. 38, no. 4, 1992, pp. 1258–1270.

16. D.L. Vail, M.M. Veloso, and J.D. Lafferty, "Conditional Random Fields for Activity Recognition," *Proc. 6th Int'l Joint Conf. Autonomous Agents and Multiagent Systems* (AAMAS 07), ACM Press, 2007, pp. 1–8.

17. S. Roweis and Z. Ghahramani, "A Unifying Review of Linear Gaussian Models," *Neural Computation*, vol. 11, no. 2, 1999, pp. 305–345.

18. D.J. Patterson et al., "Inferring High-Level Behaviour from Low-Level Sensors," *5th Int'l Conf. Ubiquitous Computing* (UbiComp 03), vol. 5, Springer, 2003, pp. 73–89.

19. J.A. Cadzow and K. Ogino, "Adaptive ARMA Spectral Estimation," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing* (ICASSP 81), vol. 6, IEEE Press, 1981, pp. 475–479.

20. R. Jursa, B. Lange, and K. Rohrig, "Advanced Wind Power Prediction with Artificial Intelligence Methods," *Artificial Intelligence in Energy Systems and Power* (AIESP 2006), ICSC Interdisciplinary Research, 2006, pp. 1–10.

21. M. Turk and A.S. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, 1991, pp. 71–86.

22. Q. Du and J.E. Fowler, "Low-Complexity Principal Component Analysis for Hyperspectral Image Compression," *J. High Performance Computing Applications*, vol. 22, no. 4, 2008, pp. 438–448.

23. A. Hyvaerinen and E. Oja, "Independent Component Analysis: Algorithms and Applications," *Neural Networks*, vol. 13, nos. 4–5, 2000, pp. 411–430.

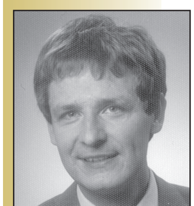
24. S. Shwartz, M. Zibulevsky, and Y.Y. Schechner, "ICA Using Kernel Entropy Estimation with nlogn Complexity," LNCS 3195, Springer, 2004, pp. 422–429.

25. N. Eagle and A.S. Pentland, "Reality Mining: Sensing Complex Social Systems," *Personal Ubiquitous Computing*, vol. 10, no. 4, 2006, pp. 255–268.

the AUTHORS

Stephan Sigg is a postdoctoral researcher in the Distributed and Ubiquitous Systems (DUS) group at the Technische Universität Braunschweig, Germany. His research interests include the design, analysis, and optimization of algorithms for ubiquitous computing environments. Sigg has a PhD in computer sciences from the University of Kassel. He's a member of IEEE, the ACM, and GI. Contact him at sigg@ibr.cs.tu-bs.de.

Sandra Haseloff is a program director at the Alexander von Humboldt Foundation. Her research interests include ubiquitous computing, mobile middleware, personalization, and context awareness. Haseloff has a PhD in computer science from Technical University of Berlin. Contact her at haseloff@web.de.



Klaus David is a full university professor and head of the Institute of Communication Technology at the University of Kassel. His research focuses on aspects of mobile middleware, personalization, and the link layer. David has a PhD in physics from the University of Siegen, Germany. He's a member of IEEE, the ACM and GI. Contact him at david@uni-kassel.de.