

Decision Making Strategies for Mobile Collaborative Context Recognition

Invited Paper

Ville Könönen
VTT Technical Research Centre of Finland
P.O. Box 1100
FI-90571 Oulu
FINLAND
ville.kononen@vtt.fi

Jani Mäntyjärvi
VTT Technical Research Centre of Finland
P.O. Box 1100
FI-90571 Oulu
FINLAND
jani.mantjarvi@vtt.fi

ABSTRACT

This paper focuses to a problem domain where several imperfect context recognition devices with restricted resources are located in the same context. The main aim of the paper is to compare different collaborative decision making methods in this domain. In particular, the focus is on certain voting mechanisms and confidence evaluation metrics. Also computational, space, and communication requirements of the discussed methods are discussed briefly. The methods are tested with a real-world dataset containing data from several sport activities. In addition, an online context recognition platform that is capable to make distributed context recognition decisions is introduced briefly.

Keywords

Context recognition, mobile devices, pattern classification, distributed decision making, voting

1. INTRODUCTION

Currently several customer products have built-in sensor units and are therefore capable to make observations about their context including information about the environment of the devices and the state of its user. Usually these devices have quite limited computational, space, and power resources. Therefore there exists an urgent need for developing light-weight context recognition methods for these devices.

The core of a context recognition system is a pattern classification algorithm that maps observations to context classes. In real-world environments, the observations are always imprecise leading to errors in the context recognition procedures. Also, the model in the pattern classification system is not perfect either, which causes additional errors in the classification. Therefore, the context recognition system is always imperfect.

If several imperfect context recognition systems are located in the same context and they can utilize information available from each other, their joint context recognition accuracy can be significantly higher than individual context recognition accuracies. There are two main reasons for that: (1) although the environment where the context recognizers are located is the same, they all observe it in a slightly different way and therefore their internal model for the pattern classification algorithm can be different. (2) if several imperfect context recognizer join their estimates in a clever way, their joint estimate can be more accurate. In this paper we focus on the second case.

The research questions motivating this paper are:

- Given that there are several mechanisms for constructing joint estimates of the context, how much advantage can be got by using these techniques compared to individual context recognizers?
- Is it tractable, by the means of computational and space resources, to compute these joint estimates?

Distributed rational decision making mechanisms such as auctions and voting protocols are extensively studied in the game theory and machine learning research fields. A good survey of different mechanisms is [5]. Using voting mechanisms as a basis for distributed pattern recognition is studied in [3]. Committee machines provide an alternative way to combine different classifiers, see e.g. [2]. Combining local context information with the information from other context recognizers is studied in [4].

The paper is organized as follows. At first, in Section 2 we introduce a light-weight pattern classification algorithm suitable for mobile devices. In Section 3, we discuss various voting mechanisms that are suitable for computing joint context recognition estimates. In Section 4, we proceed to a real-world example that is used for empirical testing of the discussed methods. In Section 5, we introduce a platform for Symbian S60 mobile phones that can be used in distributed context recognition. Finally, we conclude the paper in Section 6.

2. LIGHT-WEIGHT CLASSIFICATION SCHEME

In this section, we discuss a pattern classification algorithm that has very small computational and space requirements and is therefore especially suitable for mobile devices.

We start the section by deriving the algorithm and then we show computational requirements of the algorithm.

2.1 Minimum-distance classifier

Perhaps the simplest way to do classification of samples is to calculate a distance from a sample to the ideal elements that represent classes in the best possible way and then select the class to which the distance is the smallest. If the samples are represented in an N -dimensional vector space then there will be $N-1$ dimensional hyperplane separating each class. In that case, it would be natural to use mean value as the ideal element of each class.

More rigorously, the whole classification procedure can be expressed as follows: consider a classification task where we have to associate an N -dimensional sample \mathbf{s} to one of the C -classes. For each class $j = 1, \dots, C$, we have I^j example samples $\mathbf{x}_i^j, i = 1, \dots, I^j$. Further, let \mathbf{c}^j represent the ideal vector for the class j , i.e.

$$\mathbf{c}^j = \frac{1}{I^j} \sum_{i=1}^{I^j} \mathbf{x}_i^j.$$

Now, the classification to the class j^* can be accomplished as follows:

$$j^* = \arg \min_{j=1}^C \|\mathbf{s} - \mathbf{c}^j\|,$$

where $\|\cdot\|$ is a norm, e.g. Euclidean norm.

The above described linear classifier has several advantages. It has small computational and space requirements. It is easy to implement on various platforms and teaching the classifier is very efficient. Moreover, the classifier can be extended to be adaptive easily.

In reality the data is rarely linearly separable and therefore linear classifiers lead to the suboptimal performance. However, selecting a suitable set of features, i.e. the features that maximize linear separativity of the classes, it is possible to achieve a good classification accuracy also with linear classifiers.

2.2 Computational requirements of Minimum-distance classifier

The major advantage of the Minimum-distance classifier is its small computational and space requirements. Let us assume that we have C classes. Each of these classes is represented by an ideal vector in N -dimensional vector space. Then we have to store C N -dimensional vectors for the classifier. For each sample to be classified, we have to compare the sample with C ideal vectors to be able to select the nearest class. Usually C is quite a small number and hence computational and space requirements are small. Of course, the ideal vectors should also be calculated, which requires $\sum_{j=1}^C I^j$ additions but this can be done off-line.

3. DISTRIBUTED DECISION MAKING

Decision making is an old and mature research area. Usually the goal is to make optimal decisions with respect to a utility function modelling preferences of the decision maker. However, if there are multiple decision makers and they only maximize their own utility value, the final outcome for all decision makers can be suboptimal. There are several possible ways to model other decision makers, for example game theory, probabilistic models, etc. In this paper, we will

adopt a strategy where all the decision makers are acting individually but they also agree with a common interaction mechanism, voting mechanisms in this paper. We start the section by introducing a few voting mechanisms and then we proceed with providing some theoretical results for these mechanism. At the end of the section we situate voting in the area of context recognition and discuss communication requirements of the voting mechanisms.

3.1 Voting

In the voting mechanisms the basic setting is as follows. Several independent voters vote on the common subject. In addition, all the voters are assumed to agree with the voting mechanism. Different voting mechanisms include:

- *Borda count.* Let us assume that we have m alternatives. Each voter gives m votes to its first choice, $m-1$ votes to the second choice and proceeds similarly until the last one gets only 1 vote. Finally, the alternative with the maximal number of votes is selected to be the final outcome of the mechanism.
- *Majority voting.* Each voter gives a vote to its preferred option, the final outcome of the mechanism is the alternative with the highest number of votes.
- *Weighted majority voting.* Each voter gives a confidence value for its preferred option. Then the confidence values are summed for each alternative. The final outcome of the mechanism is the alternative with the highest summed confidence value.
- *Binary protocol.* All the options are presented to the voters pairwise. Each voter selects one of two options and the winner is compared against a new, unseen option. The process is continued until there are no new options left. The undesirable property of the binary protocol is that the final outcome of the mechanism is depended on the order in which the options are presented to the voters. Therefore we decided to leave this mechanism outside of our inspections in this paper.

In these protocols, it is assumed that all the voters make give their votes in the sincere way. If this not the case, i.e. we have a voting among *insincere* voters, the function of a voting mechanism can change in a drastic way. There is a lot of research on this are in the game theory literature, see e.g. [5].

3.2 Voting in context recognition

In the context recognition research field, a context recognition system can be seen as a voter that votes on different contexts. Although there can be several context recognizers in the same context, they all might have a slightly different model of the context and therefore they ability to select the right context is also different.

As an illustrative example, consider a simple case where we have two possible contexts and N context recognizer devices located in the same context. Let us further assume that all the context recognizers share the same probability P to select the right context class. If we apply the majority voting mechanism, we can identify two distinct cases, one with odd number of context recognizers and one with even number of recognizers. Then the total probability for selecting the right context class is given by the Eq. (1) for the

first case and Eq. (2) for the latter case. In the latter case, if both options get the equal number of votes, we select the context class randomly.

$$\sum_{i=0}^{\frac{N-1}{2}} \binom{N}{i} P^{N-i} (1-P)^i \quad (1)$$

$$\sum_{i=0}^{\frac{N-2}{2}} \binom{N}{i} P^{N-i} (1-P)^i + \frac{1}{2} \binom{N}{\frac{N}{2}} P^{\frac{N}{2}} (1-P)^{\frac{N}{2}} \quad (2)$$

The total probability the obtain the right context recognition result, defined in Eqs. (1) and (2), is plotted against the number of context recognizer in Fig. 1. It can be seen that the joint context recognition accuracy increases drastically if the individual context recognizers are reasonable good, i.e. their context recognition accuracy is more than 80%. If the context recognizers perform very badly, i.e. their accuracy is below 0.5, the majority voting decreases the joint accuracy further. The reason is that then the wrong context class has the bigger change to get selected than the right one. Fortunately, if there exist more classes, the change that the wrong context gets majority of votes is very small. This can be seen in Fig. 2 where there exists nine context classes.

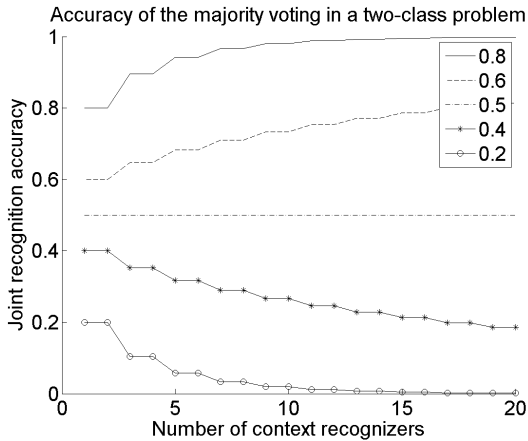


Figure 1: Two-class context recognition example.

3.3 Confidence in Majority Voting

A crucial question with the weighted majority is how to define weights for the context estimates. In context recognition systems, a core component is a pattern classification method that maps the observations to the context estimates. Therefore there should be a way to measure how well the pattern classifier is able to accomplish estimation, i.e. how well the internal model of the context recognition system is able to capture relationship between the observation and the contexts.

Such reliability information is not available from all pattern classification methods. However, the minimum-distance classifier compares the sample to be classified to the class centers and therefore we can identify the following extreme cases:

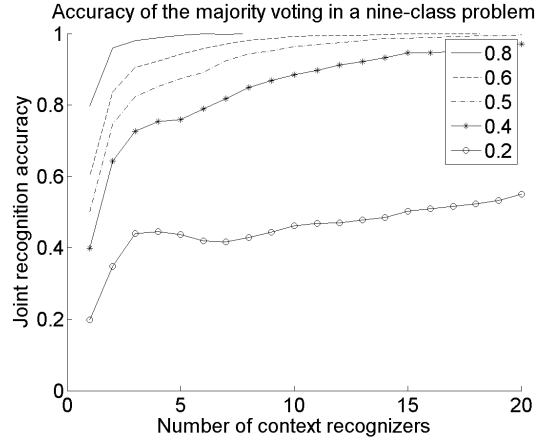


Figure 2: Nine-class context recognition example.

- A sample is located very near of a class mean vector. In this case, the sample belongs almost surely to the class.
- Distances between a sample and the mean vectors are almost identical. In this case, we cannot distinguish between the classes.

So it seems plausible that we can use the distance as a basis for the confidence evaluation. A distance between a sample s and the class mean vector c^i is shown in Eq. (3).

$$d^i = \|s - c^i\|. \quad (3)$$

It would also be a desirable property to limit the confidence values to the fixed interval, e.g. to the unit interval. This can be accomplished by dividing all the distances by the maximal distance value. The procedure is shown in Eq. (4).

$$\tilde{d}^i = \frac{d^i}{\max_{j=1 \dots C} d^j}. \quad (4)$$

In this paper, we use an absolute difference between the maximal and the minimal scaled distance value as a weighting function. The weight w for the estimate is therefore:

$$w = 1 - \min_{i=1 \dots C} \tilde{d}^i. \quad (5)$$

3.4 Communication requirements of the voting

One crucial aspect of using voting as a basis for distributed context recognition is a communication requirements of the voting mechanisms. As all the voters need to be aware of votes of other voters in the system, all context recognition estimates need to be broadcasted to all context recognizers. However, amount of broadcasted information is not large and therefore if the number of context recognizers is moderate, communication overhead stemming from voting mechanism is not remarkable.

In Borda count, each context recognizer has to order possible context classes and broadcast this ordering to other context recognizers in the system. Hence, if we have m context classes, all context recognizers should broadcast m -dimensional vectors. In the plain majority voting, each context recognizer broadcasts only its estimate to other context recognizers in the system. Correspondingly, in weighted majority voting also a confidence value is broadcasted.

4. CONTEXT RECOGNITION EXAMPLE

A realistic collaborative context recognition example is presented in this section. We start the section by introducing the Palantir-dataset, which is used for empirical testing in this paper. Then we discuss test settings in general and show numerical results from the test runs.

4.1 Dataset in empirical testing

We use the dataset described in detail in [1] as an off-line data collection for testing methods discussed in this paper. It is a large data library of realistic context information collected by using various sensors such as accelerometers and physiological sensors. The data were collected in various sport activities such as running and walking. In addition to these simple activities, also a number of combined activities were recorded, e.g. shopping, eating in the restaurant, simplified soccer playing (passing a ball between two persons) etc. In this paper we focus only on the simple activities and soccer playing.

Twelve persons took part in the data collection as test persons. Most test persons were university students, their age range was from 19 to 49 years (average 27). The whole test session for each test subject was several hours long, average almost 7 hours, and it was divided into two sections: fixed pre-defined activities and free section. During fixed activities, there was a particular person present who made annotations. During free section, persons were allowed to do freely activities they were interested in and annotations were done by test persons themselves. In this study we do not make difference between these two annotation schemes.

Sensors in the system collect both physiological and environmental information. The system has four main components:

- Embla A10 19-channel recorder device
- Garmin eTrex Venture GPS device
- Suunto X6HR wrist-top computer
- iPaq PDA

Data were stored in the PCMCIA card attached to the Embla device. In this paper, we only use acceleration and heart rate signals. Overview of the relevant component placement is depicted in Fig. 3.

The sampling frequency for acceleration sensors was 20 Hz. In addition to the raw signals directly gathered from sensors, the data library contains several features computed from the raw data. The features are calculated by windowing the corresponding raw signal with different window lengths (e.g. 10 seconds) and they include both time-domain (e.g. maximum and minimum values) and frequency domain features (e.g. power spectrum entropy). Feature values are then interpolated so that time resolution for features is 1 second.

In the tests carried out in this paper, we have selected a subset of ten features leading to the best classification accuracy from all available features by using the sequential forward selection method [6]. The selected features were based solely on the acceleration signals.

4.2 Test settings

During the data collection phase of Palantir-data library, several people collected data independently. However, the

E = ECG electrode
A = Acceleration sensors on wrist
SB = Acceleration sensors on hip

Rucksack:
REC = 19 channel recorder

Manual annotation:
P = PDA

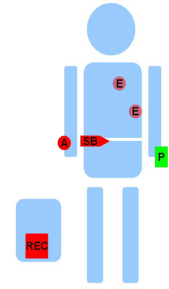


Figure 3: Overview of the sensor placement.

activities they performed were the same and the environments were almost the same. In this study we need several independent people doing the same activity at the same time and therefore we make an assumption that environments do not change between different data collection sessions and we also assume that the subjects did the activities at the same time.

In empirical tests we have a subject pool of 6 people. From this pool we select randomly a number of subjects to be placed to the same environment. A number of people varied from only 1 person to all 6 persons. The activities used in this study are enlisted in Table 1.

Table 1: Activities used in the context recognition test settings in this paper. Simplified soccer included walking, running, and passing a ball between two test subjects.

Bicycling
Simplified soccer
Lie
Nordic walking
Rowing
Running
Sitting
Standing
Walking

4.3 Empirical results

The joint context recognition accuracies are shown in Fig. 4. In all cases, the joint recognition accuracy increased with the number of context recognizers. Borda count leads clearly the worst joint accuracy although with only two recognizers it is performing better than the majority voting mechanism. The difference between the plain and the weighted version of the majority voting is that the uncertain context recognizers do not contribute significantly to the joint recognition result. In these test runs, in almost all cases, there exist at least one context recognition with very small contribution to the joint recognition result. In Table. 2, the results are shown in the numeric form.

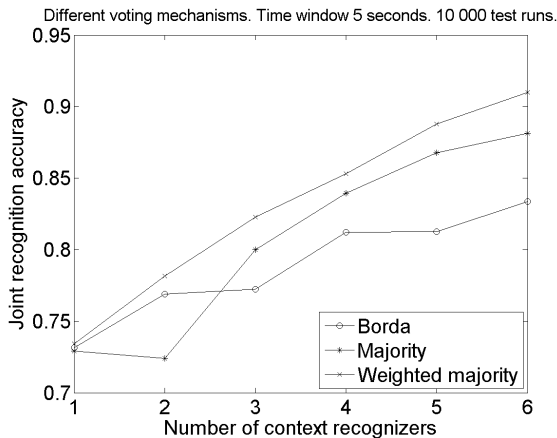


Figure 4: Different voting mechanisms in the context recognition applied to the Palantir-data.

Table 2: Context recognition accuracies with different voting mechanisms and number of voters.

	Borda	Majority	Weighted majority
1	0.73	0.73	0.73
2	0.77	0.72	0.78
3	0.77	0.80	0.82
4	0.81	0.84	0.85
5	0.81	0.87	0.89
6	0.83	0.88	0.91

5. DISTRIBUTED CONTEXT RECOGNITION PLATFORM

For testing distributed context recognition methods online we have implemented context recognition platform for Symbian S60 mobile phones. The platform includes the light-weight pattern classification method discussed in this paper and it can employ sensor data from various sensors, either built-in or via bluetooth interface. It is easy to plug in different feature calculation algorithms to the platform.

The platform is also able to send information on different levels to an external server. In the current version of the platform, the information includes raw data, calculated feature values, and context estimates. The server side is implemented by using the Apache Tomcat server. The server, in its turn, can send the information to other connected context recognizers providing necessary information for distributed decision making. As the platform sends all the relevant information for context recognition to the server, it is possible to apply more elaborated and complex data mining and pattern classification algorithms to the data. A schematic view of the system is depicted in Fig. 5. All mobile phones are able to act as clients sending their decisions and data to the server or terminals observing context estimates of other devices in the system. Data is sent to the server by using HTTP-protocol and the context estimates are sent to the mobile phones by using TCP-protocol. As all computation is performed in the mobile devices, the platform can also be extended to use some other connection type for broadcasting relevant data for distributed context recognition to its peers, for example Bluetooth connection between devices. However, service discovery times are quite long

in the current Bluetooth implementations and the range of Bluetooth communication is small limiting communication only to nearby devices.

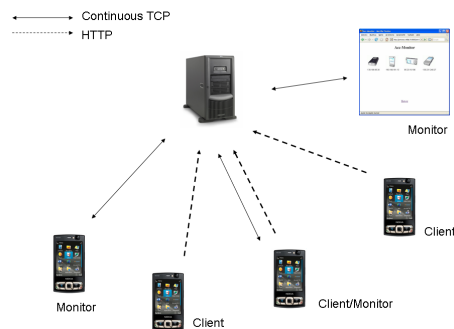


Figure 5: A schematic view of the distributed context recognition platform.

6. CONCLUSIONS

Multiple voting mechanisms were compared in the domain of mobile devices with restricted resources. The mechanisms were tested with off-line data library consisting data from sport activities. In addition, a context recognition platform form Symbian S60 mobile phones that is based on the methods discussed in this paper was introduced briefly.

The future research directions include user testing of the context recognition platform in collaborative context recognition tasks. Another interesting way to develop discussed methods further is to add adaptivity to the classifier and test how good context recognition accuracies can be get with adaptive and distributed context recognition system.

7. REFERENCES

- [1] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):20–26, 2008.
- [2] S. Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall, 2 edition, 1999.
- [3] L. Lam and S. Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transaction on Systems, Man and Cybernetics, Part A*, 27:553–568, 1997.
- [4] J. Mäntyjärvi, J. Himberg, and P. Huuskonen. Collaborative context recognition for handheld devices. In *Proceedings of the first IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, pages 161–168, 2003.
- [5] T. Sandholm. *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, chapter Distributed Rational Decision Making, pages 201–258. MIT Press, 1999.
- [6] A. Whitney. A direct method for of nonparametric measurement selection. *IEEE Transactions on Computers*, C-20(9):1100–1103, 1971.