STUDIA INFORMATICA 2016
Volume 37 Number 1 (123)

Aleksandra KARPUS
Gdansk University of Technology, Faculty of Electronics, Telecommunications & Informatics

Tommaso DI NOIA, Paolo TOMEO
Polytechnic University of Bari, Electrical & Information Engineering Department

Krzysztof GOCZYŁA
Gdansk University of Technology, Faculty of Electronics, Telecommunications & Informatics

USING CONTEXTUAL CONDITIONAL PREFERENCES FOR RECOMMENDATION TASKS: A CASE STUDY IN THE MOVIE DOMAIN

Summary. Recommendation engines aim to propose users items they are interested in by looking at the user interaction with a system. However, individual interests may be drastically influenced by the context in which decisions are taken. We present an attempt to model user interests via a set of contextual conditional preferences. We show that usage of proposed preferences gives reasonable values of the accuracy and the precision even when the dataset is quite small.

Keywords: recommendation systems, context awareness, conditional preferences

TWORZENIE REKOMENDACJI Z WYKORZYSTANIEM KONTEKSTOWYCH PREFERENCJI WARUNKOWYCH: STUDIUM PRZYPADKU W DZIEDZINIE FILMÓW

Streszczenie. Systemy rekomendacyjne sugerują użytkownikom produkty, którymi mogą być zainteresowani, na podstawie wcześniejszej interakcji z systemem. Jednak duży wpływ na decyzję użytkownika ma kontekst, w którym jest ona podejmowana. W artykule zaproponowano model zainteresowań użytkownika jako zbiór kontekstowych preferencji warunkowych i pokazano, że z ich wykorzystaniem można uzyskać dużą dokładność i precyzję rekomendacji, nawet dla małych zbiorów danych.

Słowa kluczowe: systemy rekomendacyjne, kontekst, preferencje warunkowe

1. Introduction

The main aim of a recommender system is to suggest to users items they might be interested in. We know that user interests are influenced not just by item content but also by the context in which decisions are taken [1]. In this paper we explore the linkage between an item content and the circumstances when the item is chosen in order to understand if and how contextual information, such as, e.g., *mood*, *weather* and *time of the day*, influences item selection. We combine, in a compact representation, the relations existing between the context related to the users ratings and the content associated to the items. The final model is a set of conditional preferences of the form:

$$(\gamma_1 = c_1) \wedge \ldots \wedge (\gamma_n = c_n) \mid (\alpha_1 = a_1) > (\alpha_1 = a_1') \wedge \ldots \wedge (\alpha_m = a_m) > (\alpha_m = a_m'), \quad (1)$$

with γ_i being contextual attributes and α_j content ones. The above preference is read as given the context $\gamma_1 = c_1$ and … and $\gamma_n = c_n$ I prefer a_1 over a_1' for α_1 and ... and a_m over a_m' for α_m.

We show that by using this kind of preferences we are able to predict if, in a given context, the user will like an item or not. We also discuss an influence of a selection of movie features used to compute preferences on the prediction accuracy.

The remainder of this document is as follows. In Section 2 we present an algorithm of extraction two kinds of context-aware conditional preferences, i.e. individual and general. Section 3 provides a detailed information about the used dataset. In Section 4we show experimental results. Section 5 discusses the influence of a movie features selection on the values of evaluation metrics. Related work is described in Section 6. Conclusions close the paper.

2. Preferences Extraction

User interests are strongly related to the context in which decisions are taken. We connected user preferences on movies' attributes with the context in which the movie was watched in the form of a conditional preference where the condition part of each preference always contains just contextual features.

We prepared test and training datasets for hold-out validation. In this section we will focus only on the training set. The test set will be described in the section 4.

2.1. Individual Preferences Extraction

In order to elicit preference relations we split the dataset into two parts based on the value of the ratings. We assumed that ratings with values 4 and 5 were positive and the other were negative. Both datasets we divided into smaller sets containing all the context information and one of the movie features. With such prepared data we computed context-aware individual preferences for each user with at least 5 ratings within the LDOS-CoMoDa dataset. First, we ran the Prism algorithm [3] from the WEKA library[1] (version 3.6.11) to generate rules of the form

$$(\gamma_1 = c_1) \wedge ... \wedge (\gamma_n = c_n) \mid (\alpha_1 = a_1) > (\alpha_1 = a_1'). \tag{2}$$

We then compacted user preferences with the same „conditional part" into one individual preference of desired form. If for some fixed user context the value of some content parameter was the same on both sides of preference relation then this value was marked as meaningless and not taken into consideration in this context for the user.

An example of the final form of the contextual conditional preference is shown below.

$$\begin{aligned}
& season = 3 \wedge weather = 1 \wedge time = 2 \wedge sex = 2 \wedge decision = 1 \\
& \wedge ageCat = 26 \wedge country = 2 \wedge dayType = 2 \wedge location = 1 \\
& \wedge social = 1 \wedge mood = 1 \wedge physical = 1 \wedge interaction = 1 \\
& \mid genre \in \{18\} > genre \in \{8,12,7\} \\
& \wedge director \in \{5,8\} > director \in \{3,4\} \\
& \wedge movieYear \in \{1990\} > movieYear \in \{2010\} \\
& \wedge actor \in \{1,11\} > actor \in \{7,8,10,15\}.
\end{aligned} \tag{3}$$

It means that for given context (e.g. *season* is 3 – Autumn) a user prefers *genre* with id 18 to those with 8, 12 or 7 and *directors* from clusters 5 and 8 to those from clusters 3 or 4 etc.

2.2. General Preferences Extraction

Within the dataset we had many cold users. That is, many users rated a very few number of items. Such users are generally discarded when training the model behind a recommendation algorithm as it is supposed they do not carry any relevant information. Our intuition here was that they could contribute while inferring general preferences that hold for all the users. In other words our hypothesis was that there are general trends while modeling context-aware user preferences and that cold users surely contribute in generating such trends. This is the main reason why we computed both a general set of context-aware conditional preferences and a set of context-aware individual preferences for each user with at least 5 ratings within the dataset.

The main difference in the computation of general and individual preferences is that in the first case all the ratings from the dataset were treated like they were made by one person. As a consequence, we removed many contradictory values during the merging phase.

[1]http://www.cs.waikato.ac.nz/ml/weka/

3. Dataset

We performed our experiments with the LDOS-CoMoDa[2] dataset [8]. It contains 121 users and 1232 movies. There are 55 users who ranked 5 or more items. The average number of ratings per user is 19. Values for all of the attributes are represented as a number. Unknown values are denoted by „-1”.

LDOS-CoMoDa contains user interaction with the system, e.g. the rating in a 5-star scale, basic users' information and twelve additional contextual information about the situation when the user consumed the item. Some of these pieces of information were disregarded for our intent, i.e. *end emotions* and *dominant emotions*, since they were acquired immediately after the user consumed the item and do not motivate the item choice. We also computed correlation coefficients between context related attributes. We found only two of them to be strongly correlated, i.e. *city* and *country*. For further work we chose *country* feature to achieve more general information about user preferences.

Content information about multiple item dimensions is also available within the dataset. In order to find replicable preferences in such a limited dataset, we clustered actors and directors. The process was executed by mapping each actor and director to its corresponding Wikipedia page and eventually by considering their common Wikipedia categories[3]. The number of clusters are 13 for directors and 15 for actors. The choice of those numbers is based on the calculation of the within-cluster sum of squares (*withinSS* measure from the R Stats Package, version 2.15.3), picking the number corresponding to an evident break in the distribution of the *withinSS* measure against the number of clusters.

4. Experimental Results

The test data were chosen randomly and they consist of 20% of each user ratings. Every test instance contains a user context part, an item content part and a rating that the user gave to the item.

We prepared two variants of the experiment. The first one uses both kinds of generated preferences – general and individual, while the second one uses just individual preferences. The prediction algorithm for both variants is the same. For each test instance we find the most similar preferences in terms of contextual information. In order to count similarity between a preference p and a test instance ti we used the following metric:

$$\mathrm{sim}(p,ti)=\sum_{(\gamma_i,c_i)\in p} overlap(p,ti,(\gamma_i,c_i)). \tag{4}$$

[2]The data is available at http://212.235.187.145/spletnastran/raziskave/um/comoda/comoda.php
[3]https://en.wikipedia.org/wiki/Help:Category

We also used the overlap function defined as:

$$overlap(p, ti, (\gamma_i, c_i)) = \begin{cases} 1 & (\gamma_i, c_i) \in p \wedge (\gamma_i, c_i) \in context(ti), \\ 0.5 & c_i = -1, \\ 0 & otherwise. \end{cases} \tag{5}$$

The overlap function returns 1 when we are sure that the pair (γ_i, c_i) is contained both in *p* and in the contextual attributes of the test instance *context(ti)*. When it is uncertain, i.e. when the value c_i for the dimension γ_i is equal -1 (the unknown value), it returns 0.5. Otherwise 0 is returned.

After we found preferences with the most similar context, we choose the one that has the most similar values for the content features. For this purpose we used another similarity measure and overlap function defined as:

$$\mathrm{sim}_{content}(p, ti) = \sum_{(\alpha_i, a_i) \in p} overlap_{content}(p, ti, (\alpha_i, a_i)). \tag{6}$$

$$overlap_{content}(p, ti, (\alpha_i, a_i)) = \begin{cases} 1 & (\alpha_i, a_i) \in p \wedge (\alpha_i, a_i) \in content(ti), \\ 0 & a_i = -1, \\ -1 & otherwise. \end{cases} \tag{7}$$

The overlap function used here is quite different from the one used above. In the case of movie features it is more crucial to have strict matching. This is the reason why we do not reward unknown value and why we give penalty for unmatched parameter values.

It should be noticed that we need to compare similarity of the test instance content part with both sides of the preference relation in the current preference statement.

Depending on the picked preference we could predict whether the movie is more or less preferred to be watched by the user.

To evaluate the approach we used two metrics, accuracy and precision, defined as follows:

$$Accuracy = \frac{|TPos| + |TNeg|}{|TPos| + |FPos| + |TNeg| + |FNeg|}. \tag{8}$$

$$\Pr ecision = \frac{|TPos|}{|TPos| + |FPos|}. \tag{9}$$

where *TPos* is the set of all true positive predictions, *TNeg* – true negative values, *FPos* – false positive values and *FNeg* – false negative values.

Results are shown in Table 1. We see that reasoning with both, general and individual preferences gives better results than reasoning with just individual preferences.

In the second variant of the experiments (without general preferences) we were unable to find any prediction for 35 of 417 test instances, since there is no matching with individual preferences. It means that adding common preferences not only increases accuracy and precision of prediction but also allows us to find some prediction where there is no personalized information. The main reason of this situation is the small size and the sparsity of LDOS-CoMoDa dataset. Nevertheless, the results achieved by the model which combines both kinds of preferences are satisfying and looks promising for further work.

Table 1

Accuracy and Precision values for the two variants of the experiment

Experiment	Accuracy	Precision
without general preferences	47.64	61.59
with general preferences	52.52	68.32

5. The Impact of Movie Features on Accuracy and Precision Values

Intuitively we expect that some of the movie features are more important to a user than others. Probably for different users the choice of these features will be different but importance of some of content parameters seems obvious. We assume that the usage of more important movie features could reflect with better values of accuracy and precision metrics in the task of recommendation with conditional preferences. In order to confirm our intuitions and find the most important movie feature in the LDOS-CoMoDa we ran couple of times the described experiment with some modifications to it. Every time we use a different kind of preferences (i.e. individual, general and both) and a different set of movie features. For each case we computed values of accuracy and precision metrics. The results are presented in Tables 2, 3 and 4.

Table 2

Accuracy and Precision values while removing year parameter and using just one kind of preferences

Kind of preferences	Accuracy	Precision
General	53.72	70.47
Individual	48.78	63.06

We could observe that when dealing with both kind of conditional preferences and changing the first four features in just individual ones the metrics' values remain the same (see Table 3). The reason of this behavior could be the fact that in those cases only general rules were used for the recommendation task. However, accuracy and precision values slightly change for other three parameters which are most important to the user according to our intuitions i.e. *director*, *genre* and *actor*. When we remove one of them, the metrics values decrease as we expected.

A removal of the *year* attribute gives us the biggest improvement for almost all of the cases. The exception is the usage of just individual rules (see Tables 2 and 3). In this case we have to little data to remove something and still have reasonable results.

Table 3
The dependency between Accuracy and Precision values and different movie features used in the computation of context-aware preferences. The results are computed using both kind of conditional preferences and manipulating with the movie feature for one or both of them

Removed movie feature	General preferences		Individual preferences		Both preferences	
	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Budget	53.24	68.97	52.52	68.32	53.24	68.96
Year	54.20	70.98	52.52	68.32	53.96	70.83
Language	51.32	67.51	52.52	68.32	51.32	67.51
Country	53.00	69.00	52.52	68.32	53.00	69.00
Director	52.28	67.98	52.28	68.16	52.28	68.16
Genre	53.00	67.43	52.28	67.80	52.28	66.37
Actor	49.64	66.85	50.84	67.01	47.96	65.03

When we remove more than one movie attribute and one of them is *year* we can observe better precision and accuracy values. An exception is for two sets, {*actor*, *year*} and {*actor*, *genre*}. This is because *actor* and *genre* are shown to be the most important movie features for users while choosing the movie (see Table 4). Furthermore, the removal of the *year* parameter in just general preferences shows slight improvement in comparison to the both kinds of preferences according to evaluation metrics values (see Table 3).

Table 4
The dependency between Accuracy and Precision values and different movie features used in the computation of context-aware preferences. The results are computed using both kind of conditional preferences and manipulating with movie features for both of them

Movie features							Measures	
Actor	Genre	Director	Country	Language	Year	Budget	Accuracy	Precision
x	x	x	x	x	x	x	52.52	68.32
x	x	x	x	x	x	-	53.24	68.97
x	x	x	x	x	-	x	53.96	70.83
x	x	x	x	-	x	x	51.32	67.51
x	x	x	-	x	x	x	53.00	69.00
x	x	-	x	x	x	x	52.28	68.16
x	-	x	x	x	x	x	52.28	66.37
-	x	x	x	x	x	x	47.96	65.03
x	x	x	x	x	-	-	52.76	68.66
x	x	x	x	-	x	-	50.72	67.54
x	x	x	-	x	x	-	53.24	68.25
x	x	-	x	x	x	-	52.04	67.48
x	-	x	x	x	x	-	52.52	66.52
-	x	x	x	x	x	-	46.76	62.76
x	x	x	x	-	-	x	52.42	70.27
x	x	x	-	x	-	x	53.96	70.20
x	x	-	x	x	-	x	54.20	70.77

Table 4

The dependency between Accuracy and Precision values and different movie features used in the computation of context-aware preferences. The results are computed using both kind of conditional preferences and manipulating with movie features for both of them

x	-	x	x	x	-	x	52.91	67.13
-	x	x	x	x	-	x	49.28	66.30
x	x	x	-	-	x	x	52.28	68.53
x	x	-	x	-	x	x	50.60	67.19
x	-	x	x	-	x	x	51.80	65.92
-	x	x	x	-	x	x	46.76	63.89
x	x	-	-	x	x	x	52.28	68.16
x	-	x	-	x	x	x	52.04	66.07
-	x	x	-	x	x	x	47.96	64.86
x	-	-	x	x	x	x	53.00	67.12
-	x	-	x	x	x	x	48.68	64.50
-	-	x	x	x	x	x	46.28	63.79
x	x	x	x	-	-	-	50.85	68.68
x	x	x	-	x	-	-	53.24	68.78
x	x	x	-	-	x	-	52.17	68.00
x	x	x	-	-	-	x	53.28	70.53
x	x	x	-	-	-	-	51.83	68.23

6. Related Work

Besides well-known and widely-used collaborative and content-based recommendation techniques there exist also knowledge-based ones whose depend on detailed knowledge about items [7]. In this section we focus on two types of knowledge-based systems, i.e. rule-based (RBR) and case-based (CBR) reasoning.

In RBR systems the knowledge about items and users' interests is represented in the form of „IF condition THEN action” rules and new problems are answered by reasoning with them. In the recommendation task, when some condition holds the matching rule is fired [6].

CBR systems store knowledge in the casebase in the form of *cases*. During recommendation task, the cases are compared to user requirements according to some similarity metric. The items suggested by the most (least) similar cases are then tested for success by active user. The process has many iterations and all of them are kept in the casebase as new cases [9].

The recommendation technique proposed in this paper is something in between RBR and CBR techniques. Contextual conditional preferences could be seen as both, rules or cases, but in fact they are none of them. We chose active preferences according to two similarity metrics so we could position our work in the CBR research area. However, we do not have iterations or a relevance verification in the recommendation process.

The idea of modeling user interests with a preference relation is not new. In [2] a formalism of CP-nets was proposed. CP-nets are intuitive graphical models for representing conditional preferences under *ceteris paribus* („all else being equal") assumptions. They are represented as directed graphs where connections between nodes depict the dependencies between variables represented by these nodes. A conditional preference table (CP table) is associated with every node. This table is nothing more like description of conditional preference statements. Examples of this kind of preference statements are: $fantasy \succ romance$ and $fantasy \mid A.Sapkowski \succ J.R.R.Tolkien$. It should be noticed that above statements could not occur in the same CP table.

The kind of preferences presented in this paper is quite different to those described above. Proposed preferences always contain „conditional part" which consists of contextual parameters only. Another important difference is the lack of *ceteris paribus* assumption.

In [10] similar approach was presented. Authors also proposed preferences that depend on the context. But in general, this work differs from ours in some aspects. Firstly, authors used just two contextual variables, i.e. weather and location. Secondly, the preferences are not conditional and they use a score to express user interests in the item in contradiction to the preference relation used by us. Moreover, the approach presented by authors focuses on database management systems (DBMS) and uses the OLAP techniques for processing context-aware queries. DBMS and OLAP are out of our interests.

Adding contextual data, i.e. *time*, to the session-based collaborative filtering (SSCF) has been proven to give 200% better accuracy in the music domain, according to [4]. Authors extend existing SSCF algorithm by creating a feature vector which consists of 5 properties: *time of the day*, *weekday*, *day of month*, *month* and *diversity*. In our approach we use 3 temporal features: *time of the day*, *day type* and *season*, so just one property overlap. We also propose own method for making recommendation, we did not adopt any of existing one because of the new representation of a user profile.

In [5] a completely different approach is presented. Authors propose a hierarchical hidden Markov model for capturing changes in user's interest. Using this model, it is possible to predict the context of a next user's interaction with the system based on the possibility of transition between different contextual states. Predicted context is used for making recommendations. Authors shown that the usage of a hierarchical hidden Markov model increases the diversity of recommendations.

7. Conclusions

In this paper we introduced the new model for representing user preferences, i.e. contextual conditional preferences, and presented some experiments on the usage of it when dealing with a small dataset. We showed that such preferences are an interesting tool for recommendation tasks. Because of the small size and the sparsity of the LDOS-CoMoDa dataset we were unable to build a reasonable recommendation model that is fully personalized. Nevertheless, the general model looks promising and we are planning to extend our work for other datasets.

Usage of proposed context-aware conditional preferences confirmed our intuition about movie features that influence the most a users choice. Moreover, we showed that an importance of users interests in this features is naturally reflected in the values of accuracy and precision metrics. This suggests interesting directions to extend the model with weighting the movie features in contextual conditional preferences.

BIBLIOGRAPHY

1. Adomavicius G., Tuzhilin A.: Context-aware recommender systems. [in:] Ricci F., Rokach L., Shapira B., Kantor P.B. (eds.): Handbook on Recommender Systems, Springer, 2011, p. 217÷256.
2. Boutilier C., Brafman R.I., Domshlak C., Hoos H.H., Poole D.: Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. Journal of Artificial Intelligence Research, Vol. 21, 2004, p. 135÷191.
3. Cendrowska J.: PRISM: an algorithm for inducing modular rules. International Journal of Man-Machine Studies, Vol. 27, No. 4, 1987, p. 349÷370.
4. Dias R., Fonseca M.J.: Improving Music Recommendation in Session-Based Collaborative Filtering by Using Temporal Context. [in:] Proc. of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI '13), IEEE Computer Society, Washington, DC, USA 2013, p. 783÷788.
5. Hosseinzadeh Aghdam M., Hariri N., Mobasher B., Burke R.: Adapting Recommendations to Contextual Changes Using Hierarchical Hidden Markov Models. [in:] Proceedings of the 9th ACM Conference on Recommender Systems (RecSys '15), ACM, New York, NY, USA 2015, p. 241÷244.

6. Imran H., Belghis-Zadeh M., Chang T.W., Kinshuk, Graf S.: A Rule-Based Recommender System to Suggest Learning Tasks. [in:] Proc. of 12th International Conference on Intelligent Tutoring Systems (ITS 2014), LNCS, Vol. 8474, Springer, 2014, p. 672÷673.
7. Jannach D., Zanker M., Felfering A., Friedrich G.: Knowledge-based recommendation. [in:] Recommender Systems: an Introduction, Cambridge University Press, 2011, p. 81÷123.
8. Kosir A., Odic A., Kunaver M., Tkalcic M., Tasic J.F.: Database for contextual personalization. Elektrotehniski vestnik [English print ed.], Vol. 78, No. 5, 2011, p. 270÷274.
9. Smyth B.: Case-Based Recommendation. [in:] Brusilovsky P., Kobsa A., Nejdl W. (eds.): The Adaptive Web, LNCS, Vol. 4321, Springer, 2007, p. 342÷376.
10. Stefanidis K., Pitoura E., and Vassiliadis P.: A context-aware preference database system. International Journal on Pervasive Computing and Communications, Vol. 3, No. 4, 2007, p. 439÷460.

Omówienie

Systemy rekomendacyjne są obecne w naszym codziennym życiu, przez co cieszą się rosnącym zainteresowaniem zarówno naukowców, jak i przedstawicieli przemysłu. Ze względu na ich specyfikę coraz większą uwagę poświęca się kontekstowi i jego wpływowi na zadowolenie użytkownika z otrzymanych rekomendacji.

Niniejszy artykuł prezentuje nowy, zależny od kontekstu model preferencji użytkownika w formie zbioru kontekstowych preferencji warunkowych przedstawionych za pomocą wzoru (1). Do eksperymentów wykorzystano zbiór LDOS-CoMoDa w dziedzinie filmów, mający ocenę użytkownika w skali 1-5, informacje na temat filmu (*gatunek*, *reżyser* itp.) oraz kontekstu, w jakim użytkownik obejrzał dany film (*towarzystwo*, *pogoda* itp.). Założono, że oceny 4-5 są pozytywne, zaś 1-3 negatywne, otrzymując dwie klasy. Dokonano podziału zbioru uczącego na mniejsze zbiory, zachowując zawsze wszystkie dane kontekstowe i tylko jeden atrybut filmu (np. *gatunek*) oraz tylko jedną klasę ocen dla rozpatrywanego użytkownika. Na tak przygotowanych danych, w celu ekstrakcji preferencji uruchomiono algorytm Prism z biblioteki WEKA (wersja 3.6.11) i otrzymano pośrednie reguły opisane za pomocą wzoru (2). Następnie dokonano połączenia reguł mających dokładnie taki sam kontekst i wykluczenia tych sprzecznych. Przykład kontekstowej preferencji warunkowej przedstawia wzór (3). Oprócz indywidualnych preferencji wyznaczono również ogólne, przedstawiające trend zainteresowań wśród użytkowników. Różnica w procesie ekstrakcji polega na tym, że wykorzystano wszystkie dane, tak jakby należały do jednego użytkownika.

Zbiór testowy został przygotowany w sposób losowy i zawiera 20% ocen każdego użytkownika, który ocenił więcej niż 5 filmów. Do rekomendacji użyto odpowiednich par funkcji (4) i (5) oraz (6) i (7) w celu ustalenia podobieństwa pomiędzy kontekstem oraz atrybutami preferencji użytkownika i filmu ze zbioru testowego. Do oceny modelu wykorzystano dwie popularne miary, tj. dokładność i precyzję, dane za pomocą wzorów (8) i (9). Wyniki przestawiono w tabeli 1.

Dokonano również analizy wpływu doboru atrybutów filmu przy procesie ekstrakcji preferencji na otrzymane w procesie oceny rekomendacji wartości użytych miar dokładności i precyzji. Wyniki zestawiono w tabelach 2, 3 i 4.

Addresses

Aleksandra KARPUS: Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, ul. G. Narutowicza 11/12, 80-233 Gdańsk, Poland, aleksandra.karpus@eti.pg.gda.pl.

Tommaso DI NOIA: Polytechnic University of Bari, Electrical & Information Engineering Department, via E. Orabona 4, 70125 Bari, Italy, tommaso.dinoia@poliba.it.

Paolo TOMEO: Polytechnic University of Bari, Electrical & Information Engineering Department, via E. Orabona 4, 70125 Bari, Italy, paolo.tomeo@poliba.it.

Krzysztof GOCZYŁA: Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, ul. G. Narutowicza 11/12, 80-233 Gdańsk, Poland, Krzysztof.Goczyla@eti.pg.gda.pl.