

# Tensor-Based Modeling of Temporal Features for Big Data CTR Estimation

Andrzej Szwabe, Pawel Misiorek<sup>(✉)</sup>, and Michal Ciesielczyk

Institute of Control and Information Engineering, Poznan University of Technology,  
ul. Piotrowo 3a, 60-965 Poznan, Poland

{andrzej.szwabe,pawel.misiorek,michal.ciesielczyk}@put.poznan.pl

**Abstract.** In this paper we propose a simple tensor-based approach to temporal features modeling that is applicable as means for logistic regression (LR) enhancement. We evaluate experimentally the performance of an LR system based on the proposed model in the Click-Through Rate (CTR) estimation scenario involving processing of very large multi-attribute data streams. We compare our approach to the existing approaches to temporal features modeling from the perspective of the Real-Time Bidding (RTB) CTR estimation scenario. On the basis of an extensive experimental evaluation, we demonstrate that the proposed approach enables achieving an improvement of the quality of CTR estimation. We show this improvement in a Big Data application scenario of the Web user feedback prediction realized within an RTB Demand-Side Platform.

**Keywords:** Big data · Multidimensional data modeling · Context-aware recommendation · Data extraction · Data mining · Logistic regression · Click-through rate estimation · WWW · Real-Time Bidding

## 1 Introduction

Web content utility maximization is one of the main paradigms of the so-called Adaptive Web [3]. Many researchers agree that Click-Through Rate (CTR) estimation is important for maximization of Web content utility and that machine learning plays a central role in computing the expected utility of a candidate content item to a Web user. The click prediction – widely referred to as CTR estimation – is an interesting and important data mining application scenario, especially when realized on the Web scale [4, 9, 16]. Real-Time Bidding (RTB) belongs to the best examples of widely-used Big Data technologies [15]. As confirmed by many authors, the research on RTB algorithms involves facing many challenges that are typical for Big Data. In particular, RTB algorithms must be capable to process heterogeneous and very sparse multi-attribute data streams having the volume order of terabytes rather than gigabytes [4]. Moreover, to be applicable in a real-world environment, an RTB optimization algorithm must be able to provide its results in tens of milliseconds [15].

Digital advertising is a rapidly growing industry already worth billions of dollars. RTB is one of the leading sectors of the digital advertising industry. In this paper, we contribute to the intensively investigated area of research on machine learning algorithms optimizing RTB-based dynamic allocation of ads. The proposed solution faces the challenges imposed by the RTB protocol requirements and, at the same time, introduces the temporal feature engineering based on the tensor model what has not been investigated yet.

Some of the tensor-based approaches to data modeling have already been identified as addressing the Big Data challenges [5, 10]. Although the area of the research on tensor-based Big Data modeling has emerged quite recently [5], the results achieved so far, indicate that, at least in online advertising application scenarios, tensor-based approaches are able to outperform many alternative ones, including those based on the matrix factorization and deep learning [16, 21].

## 2 Related Work

As far as Big Data application scenarios are concerned, it is widely agreed that Logistic Regression (LR) is the state-of-the-art CTR estimation method [4, 16, 23]. For this reason, the scope of the research presented in this paper is limited to feature modeling applicable to a data mining system based on LR.

In the context of the RTB Demand Side Platform (DSP) optimization scenario, it is important to make the CTR estimation algorithm highly contextual and capable to exploit various data augmentations [16, 22]. In the relevant papers these two requirements are sometimes integrated into the single, more general requirement. Specifically, recommender systems deployed to perform CTR estimation are required to model the heterogeneous data attributes explicitly from multiple alternative and complementary ‘aspects’ [16]. The idea of such ‘multi-aspect’ data modeling is familiar to researchers working on tensor-based data representation methods [2, 13]. The need for ‘multi-aspect’ data modeling has been recognized by the authors of tensor-based RTB CTR estimation systems [16] and by the authors of advanced classification systems theoretically-grounded on the rough set theory [12]. All these types of data mining systems perform some type of ‘multi-aspect’ data modeling by using combinations of multiple ‘interacting’ features [4, 16].

There are a few approaches to building feature conjunctions that have been presented in the literature on RTB CTR estimation [4, 9, 16, 21]. Some of the papers involve the explicit use of the cartesian product or the tensor product in the models’ definitions [4, 16].

It is worth recalling that the tensor space is a space formed over a cartesian product of the constituent vector spaces. In the context of an algebraic feature representation, it is a straightforward and widely-followed assumption to represent features in their vector spaces and to map the feature values to the dimensions of these spaces [8, 14, 18, 20]. Although not all the authors of such feature conjunctions models explicitly refer to the tensor product as the means for building the algebraic representations of feature conjunctions, such a tensor-based definition is a direct consequence of the assumption that the constituent

features are represented by vector spaces. On the other hand, the authors of many papers presenting tensor-based models of the data sets elements having the form of the properties' conjunctions, explicitly refer to the tensor product as the means for building algebraic multi-feature data representations [8, 14, 16, 18, 20].

To the best of our knowledge, there are no publications presenting multi-linear temporal feature models designed for CTR estimation systems based on LR. Although temporal features are included in the overall feature sets of the models proposed by the authors of the leading CTR estimation algorithms based on LR [4, 9, 16, 21], none of these models is both tensor-based and used for multi-linear representation of temporal features. In consequence, none of these models involves the use of feature conjunctions of different arity [4, 9, 16]. Moreover, the impact of an application of temporal features (with or without their conjunctions) on the quality of CTR estimation has not been presented in any of the above-recalled papers. Therefore, we believe that the research results presented herein are not only practically useful, but may also be regarded as original and interesting theoretical contribution to the field of the research on feature models for CTR estimation systems.

### 3 Tensor-Based Feature Modeling

Tensor-based data modeling is a broad topic – typically investigated from the perspective of various approaches to tensor-based data processing [5, 11, 20]. It has to be stressed that the scope of the tensor-based modeling that is represented by the model proposed in this paper is relatively narrow – it is limited to (i) the ‘feature addressing’ scheme based on the tensor product of the feature-indexing standard basis vectors and (ii) the use of a simple multi-tensor network. In such a simplified form, a tensor-based model of additional features (herein referred to as metafeatures) is equivalent to the state-of-the-art feature models defined with the use of the cartesian product that are presented in the literature on CTR estimation based on LR [4, 16]. It also has the most distinctive property of any tensor-based data representation, which is the ability to represent data in its natural form in which vector space dimensions represent feature values, rather than data/training examples [14, 20]. Thanks to this property, any combination of the features may be mapped on its dedicated tensor entry. Moreover, the use of the multi-tensor hierarchy network (presented in Sect. 3.2) provides simple means for the mapping between a conjunction features' subset and the corresponding tensor network node; the arity of the conjunction tuple maps to the level of the tensor network – the level including the tensors of the order equal to the arity of the conjunction tuple.

#### 3.1 Tensor-Based Multidimensional Data Modeling

Let us use the notation in which  $\mathcal{A}, \mathcal{B}, \dots$  denote sets,  $\mathbf{A}, \mathbf{B}, \dots$  denote tensors and  $a, b, \dots$  denote scalars. The tensor-based feature model represents the multi-attribute data describing the given user feedback event (in the case of the

application scenario presented in this paper – the event representing a user click on a given ad in result of a given impression). This data have the form of a set of logs  $\mathcal{E}$ , in which each of the log entries is defined as a tuple of multiple features. We model this set as an  $n$ -order tensor:

$$\mathbf{T} = [t_{i_1, \dots, i_n}]_{m_1 \times \dots \times m_n},$$

defined in a tensor space  $\mathcal{I}_1 \otimes \dots \otimes \mathcal{I}_n$ , where each  $\mathcal{I}_i$ ,  $1 \leq i \leq n$  indicates a standard basis [14] of dimension  $|\mathcal{I}_i| = m_i$  used to index elements of the domain  $\mathcal{F}_i$  – the domain of feature  $i$ . The entries of tensor  $t_{i_1, \dots, i_n}$  represent the outcome of the investigated event – formally described using the function  $\psi : \mathcal{F}_1 \times \dots \times \mathcal{F}_n \rightarrow \mathbb{R}$ . In the case of RTB CTR prediction task, the events' outcomes are usually described by means of the binary-valued function  $\psi : \mathcal{F}_1 \times \dots \times \mathcal{F}_n \rightarrow \{0, 1\}$  defining *non-click* and *click* events, respectively.

Since the input data form a sparse (incomplete) multidimensional structure, the tensor  $\mathbf{T}$  is usually stored in the form of  $n$ -tuples, for which a given tuple  $\gamma$  is modeled as:

$$\gamma = (w^\gamma, f_1^\gamma, \dots, f_n^\gamma),$$

where  $f_i^\gamma \in \mathcal{F}_i$  are the feature values defining the tuple  $\gamma$  and  $w^\gamma = \psi(f_1^\gamma, \dots, f_n^\gamma)$  denotes its weight.

### 3.2 Multi-Tensor Hierarchy Network

In this paper we used the model, referred to as *Multi-Tensor Hierarchy Network* (MTHN), enabling the representation of correlations observed in any subset of the feature set. In contrast to other approaches (e.g., [16]), we do not use any heuristic method for feature grouping which is necessary when simplifying the model, e.g., to the single third-order tensor. The proposed model provides the averaging framework enabling to represent the means within a network of tensors, which is used for combinatorial exploration of all the possible subsets of the features.

Let  $[n] = \{1, 2, \dots, n\}$  denotes the set enumerating features describing the investigated event. For each subset  $\mathcal{S} = \{p_1, \dots, p_k\} \subset [n]$  we construct the tensor  $\mathbf{T}(\mathcal{S})$  by averaging the data throughout all non-missing (i.e., known) values in the respective  $(n - k)$ -dimensional 'sub-tensor' of the input  $n$ -order tensor  $\mathbf{T}$ . The fibres (one-dimensional fragments of a tensor, obtained by fixing all indices but one) and slices (two-dimensional fragments of a tensor, each being obtained by fixing all indices but two) are the examples of such sub-tensors that are most commonly referenced in the literature [11, 13].

Formally, for each subset  $\mathcal{S} = \{p_1, \dots, p_k\}$  of  $[n]$ , where  $0 \leq k < n$ , such that  $\mathcal{R} = [n] \setminus \mathcal{S} = \{r_1, \dots, r_{n-k}\}$ , we construct tensor  $\mathbf{T}(\mathcal{S}) = [t(\mathcal{S})_{j_1, \dots, j_k}]_{m_{p_1} \times \dots \times m_{p_k}}$  in such a way that, for a given combination of feature values in  $\mathcal{S}$ , we have:

$$t(\mathcal{S})_{j_1, \dots, j_k} = \frac{1}{z} \sum_{i_{r_1}=1}^{m_{r_1}} \dots \sum_{i_{r_{n-k}}=1}^{m_{r_{n-k}}} \sum_{i_{p_1}=j_1}^{j_1} \dots \sum_{i_{p_k}=j_k}^{j_k} t_{i_1, \dots, i_n} \quad (1)$$

if  $z > 0$ , where  $z$  is a number of all known (i.e., non-missing) values in a sub-tensor of tensor  $\mathbf{T}$  defined by fixing  $i_{p_1}, \dots, i_{p_k}$  to feature values indices  $j_1, \dots, j_k$ . The value  $t(\mathcal{S})_{j_1, \dots, j_k}$  may be equivalently seen as the weight  $w^\phi$  of the ‘shortened’ tuple  $\phi = (w^\phi, f_{p_1}^\phi, \dots, f_{p_k}^\phi)$ . Note that in the case of RTB CTR modeling, value  $w^\phi$  is just the CTR observed over all the events for which the features  $p_1, \dots, p_k$  have their values equal to  $j_1, \dots, j_k$ .

Tensors  $\mathbf{T}(\mathcal{S})$  form the hierarchical network of  $2^n$  tensor structures of orders from the set  $\{0, \dots, n\}$  – called Multi-Tensor Hierarchy Network – corresponding to all possible subsets of the set of  $n$  investigated features. In particular, the model consist of:

- level 0 of MTHN – containing one node which is the tensor of order 0  $\mathbf{T}(\emptyset)$  – the scalar representing the weight value averaged over all known events (the averaged CTR in RTB CTR estimation application scenario),
- for  $k \in \{1, \dots, n-1\}$ : level  $k$  of MTHN – containing  $\binom{n}{k}$   $k$ -order tensors  $\mathbf{T}(\{p_1, \dots, p_k\})$  representing the averages over events with  $k$  features with fixed values,
- level  $n$  of MTHN – containing one node –  $n$ -order tensor  $\mathbf{T}([n]) = \mathbf{T}$  storing the weights of input  $n$ -tuples.

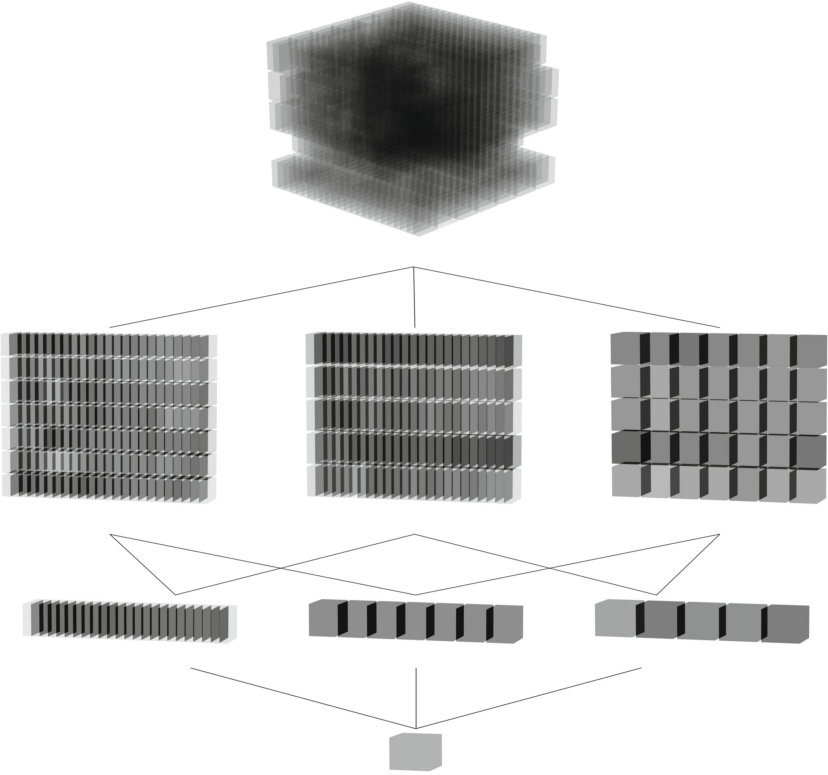
Two nodes  $\mathbf{T}(\mathcal{S})$  and  $\mathbf{T}(\mathcal{S}')$  of MTHN are connected if and only if they belong to neighboring levels ( $\|\mathcal{S}|-|\mathcal{S}'\|=1$ ) and the set of features modeled by one of them is a subset of features modeled by the other ( $|\mathcal{S} \cap \mathcal{S}'| = |\mathcal{S}|$  or  $|\mathcal{S} \cap \mathcal{S}'| = |\mathcal{S}'|$ ) – in other words – one of them may be obtained from the other by averaging over a single tensor mode.

The example of hierarchy of tensors  $\mathbf{T}(\mathcal{S})$  for set  $\mathcal{S} \subset [3] = \{1, 2, 3\}$  corresponding to features ‘hour’, ‘weekday’ and ‘advertiser’, respectively, is illustrated in Fig. 1.

## 4 Modeling Temporal Features Using MTHN

Although in this paper we investigate temporal features modeling, the MTHN metamodel presented in Sect. 3 may be used to represent data tuples defined on the basis of any feature sets, not only temporal ones. To illustrate the modeling of temporal features performed in accordance with the approach proposed herein, let us present just one of many possible MTHN metamodel use cases. In Fig. 1 an order-three MTHN-based model is presented that exemplifies the case of the MTHN-based model built to represent average CTR values that reflect jointly the content of the advertisers set (five advertisers for the 2nd season of the iPinYou dataset) and two temporal features sets – the set consisted of the one-hour long nychthemeron (day and night) time slots (twenty four hours of the day) and the set of seven weekdays. This case is one of the cases that we evaluated experimentally in Sect. 6.

To build the MTHN metamodel visualization presented in Fig. 1, we used the set  $\mathcal{S} = [3] = \{1, 2, 3\}$  to enumerate three features as follows: index 1 was used to represent 24 one-hour long nychthemeron time-slots, index 2 was used



**Fig. 1.** Visualization of a MTHN-based representation of the average CTR values in the case of the 2nd season of the iPinYou dataset, with the advertisers set and two temporal features sets consisted of the one-hour time slots set and the weekdays set.

to represent 7 weekdays, and index 3 was used to represent 5 advertisers of the iPinYou dataset Season 2 [23]. The Fig. 1 presents MTHN consisting of 4 levels enumerated from bottom to top. In particular, level 0 is just tensor  $\mathbf{T}(\emptyset)$ . Level 1 contains tensors  $\mathbf{T}(\{1\}), \mathbf{T}(\{2\}), \mathbf{T}(\{3\})$  (presented from left to right) representing features values corresponding to hours, weekdays and advertisers, respectively. Level 2 contains tensors  $\mathbf{T}(\{1, 2\}), \mathbf{T}(\{1, 3\}), \mathbf{T}(\{2, 3\})$  representing feature conjunctions of arity 2 of the form ‘hour  $\times$  weekday’, ‘hour  $\times$  advertiser’, and ‘weekday  $\times$  advertiser’ respectively. Finally, level 3 contains the tensor  $\mathbf{T}([n])$  representing the feature conjunctions of arity 3 of the form ‘hour  $\times$  weekday  $\times$  advertiser’.

In Fig. 1 the darkness of each box – representing a given entry of the given tensor (in cases of some of the MTHN nodes, being a special case of a tensor: a vector or a scalar) – illustrates the average CTR value observed for the feature conjunction corresponding to this entry (the darker the box the higher the CTR value). It should be stressed that such a visualization is just a demonstration of an example of MTHN data structure application. In particular, the visualized

CTR values should not be confused with values of LR feature weights. On the other hand, in the particular case of average CTR values visualization, a MTHN data structure of the proposed kind may be regarded as an novel ‘analytics tool’ supporting a researcher in his/her analysis of different average CTR values – observed contextually for different sets of features used for the tensor-based ‘segmentation’ of the features.

## 5 Evaluation Methodology

The main goal of the experimentation reported in this paper was to evaluate the impact of the selected cases of an application of MTHN-based temporal feature models on the RTB CTR estimation quality.

The analysis presented herein is based on extensive offline experiments involving the use of the iPinYou dataset and selected CTR estimation quality measures. We have assumed that the ultimate goal of our experimentation reported herein is to present experimentally confirmed findings that are straightforwardly applicable to RTB CTR estimation systems based on the LR framework.

### 5.1 Measures

Most of the authors of papers on RTB CTR methods presenting results of offline experiments use the Area Underneath the ROC curve (AuROC) metric [4, 16, 21, 23], regarding it as enabling one to directly evaluate the systems’ ability to distinguish between accurate and inaccurate predictions [7].

Nonetheless, AuROC, despite being useful for heavy-tailed recommendation or link prediction systems [6, 17, 19], may not provide a full insight into the RTB CTR estimation problem. Taking into account both the popularity and the limitations of AuROC, complementary to the presentation of our AuROC results, we show the Average Precision (AP) results (equivalent to the area underneath the precision-recall curve). While both metrics measure true positive rate, AP emphasizes precision while AuROC emphasizes false positive rate [6, 17]. Such a difference of how the true negatives are treated is especially evident when the number of negative observations (*non-clicks*) is significantly higher than the positive ones (*clicks*).

As realized by some authors [22, 23], in the context of the RTB, the quality of CTR estimation should be measured in a way that reflects the real-world requirement of the system’s ability to preserve a high Key Performance Index value under the time constraints of the given campaign execution. In terms of precision and recall measures, a useful RTB optimization cannot severely reduce the bidding frequency. This means that the increase of precision should not lead to a severe reduction of recall. Under such conditions, the CTR estimation results presented as ‘summarized’ curves – such as AuROC and AP – are not considered as sufficiently informative for a real-world DSP.

Following the above-stated observation, in the analysis of experiments presented in Sect. 6 we additionally analyze the results from the perspective of the

CTR estimation system’s ability of achieving different trade-offs between precision (i.e., CTR) and recall.

## 5.2 Dataset

To evaluate the proposed model we used the first publicly available large-scale RTB dataset released by iPinYou Information Technologies Co., Ltd [23]. The dataset contains impression, click, and conversion logs collected from several campaigns of different advertisers during various days and is divided into a training set and a test set. Each record contains five types of information: (i) temporal features (timestamp of the bid request), (ii) user features (iPinYou ID, browser user-agent, IP address, etc.), (iii) ad features (creative ID, advertiser ID, landing page, etc.), (iv) publisher features (domain, URL, ad slot ID, size, visibility, etc.), and (v) other features regarding the RTB auction (bid ID, bidding price, winning price, etc.).

In this paper we partition the dataset in two different ways:

- (a) by timestamps, in the same manner as described in [16, 23],
- (b) randomly, using the same training ratios ( $tr$ ) as in (a) (i.e.,  $tr \approx 0.7897$  for season 2 and  $tr \approx 0.6667$  for season 3).

The major dataset statistics are shown in Table 1. More detailed information on the dataset may be found in [16, 23].

**Table 1.** Dataset statistics (using the partitioning by timestamps).

Season	Dataset	Impressions	Number of feature values	Clicks	CTR (%)
2	Training set	12, 190, 438	801,890	8,838	0.073
	Test set	2, 521, 627	543,711	1,873	0.074
3	Training set	3, 147, 801	589,872	2,700	0.086
	Test set	1, 579, 071	482,208	1,135	0.072

## 6 Experiments

We trained the CTR estimator to predict the probability of the user click on a given ad impression using information extraction from raw user feedback data. As suggested in [23], in each of the tested variants the following pre-processing was performed:

- The timestamps were generalized into the corresponding weekday and hour value.
- The OS and browser names were extracted from the user-agent field.
- The floor prices were quantized into the buckets of 0, [1, 10], [11, 50], [51, 100] and [101,  $+\infty$ ).

**Table 2.** CTR estimation performance in terms of AuROC and AP (%); the best results in each row are highlighted by bold font setting.

Dataset partitioning	iPinYou dataset season	<i>day+hour</i>		<i>MTHN(domain)</i>		<i>MTHN(advertiser)</i>	
		AuROC	AP	AuROC	AP	AuROC	AP
By timestamp	2	<b>90.67</b>	15.580	<b>90.67</b>	15.591	90.54	<b>16.243</b>
	3	75.23	0.312	<b>76.52</b>	<b>0.328</b>	76.01	0.317
Random	2	86.19	11.671	<b>86.52</b>	11.781	85.84	<b>11.922</b>
	3	78.91	0.530	<b>79.22</b>	<b>0.555</b>	78.66	0.531

We evaluated two variants based on the proposed tensor-based feature modeling metamodel:

- *MTHN(domain)* – reflecting two temporal features sets consisted of weekdays and hours, and the content of the domains set (the set of Web domains offering ad impressions),
- *MTHN(advertiser)* – reflecting two temporal features sets consisted of weekdays and hours, and the content of the advertisers set.

The state-of-the-art algorithm based on the basic temporal feature modeling proposed in [23] (referred to as *day+hour*) was chosen as a baseline.

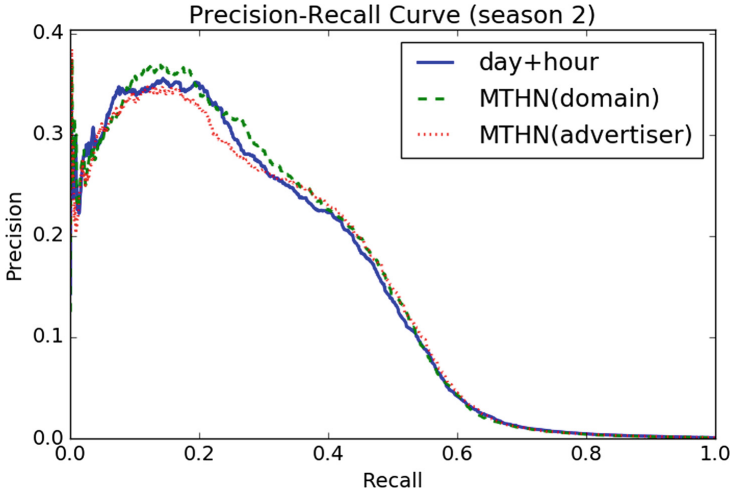
To learn the LR model parameters, we used the Stochastic Gradient Descent (SGD) algorithm. The initial parameters (i.e., weights corresponding to the binary feature values) were set to 0. The learning rate in all the experiments was set to 0.01. The tolerance for the stopping criterion was set to 0.0001. Specifically, the learning was stopped when the logistic loss value change observed between two consecutive iterations reached the specified tolerance-defining threshold value. The training examples were randomly shuffled after each iteration so as not to introduce a bias into the optimization results [1].

The CTR estimation performance results concerning both dataset partitioning scenarios – the timestamp-based one and the random one – are presented in Tables 2 and 3. In the case of the random approach, the mean values from 10 experiments are shown for all the presented measures. Table 2 demonstrates the performance comparison provided by means of AuROC and AP measures. The standard error of each presented mean is less than 0.1% and 0.02% for AuROC and AP correspondingly.

Table 3 presents the CTR estimation system’s ability of achieving different trade-offs between precision (i.e., CTR) and recall. Specifically, CTR values for recall equal to  $1/8$  and  $1/4$  were evaluated. Finally, Fig. 2 presents the precision vs recall curve (i.e., all CTR values) for iPinYou Dataset Season 2.

**Table 3.** CTR estimation performance in terms of  $P(R=1/8)$  and  $P(R=1/4)$  (%); the best results in each row are highlighted by bold font setting.

Dataset partitioning	iPinYou Dataset season	<i>day+hour</i>		<i>MTHN(domain)</i>		<i>MTHN(advertiser)</i>	
		$P(R=1/8)$	$P(R=1/4)$	$P(R=1/8)$	$P(R=1/4)$	$P(R=1/8)$	$P(R=1/4)$
By timestamp	2	33.62	29.46	<b>38.91</b>	<b>31.73</b>	34.41	32.37
	3	<b>0.58</b>	0.39	0.54	<b>0.44</b>	0.52	0.42
Random	2	33.16	23.11	<b>33.74</b>	<b>24.05</b>	31.58	23.16
	3	1.01	0.68	<b>1.08</b>	<b>0.73</b>	1.03	0.72

**Fig. 2.** PvR curve for Season 2 partitioned by timestamps.

## 7 Conclusions

On the basis of the extensive experimental evaluation (presented in Sect. 6), we have demonstrated that the proposed tensor-based model of temporal features enables to improve the quality of CTR estimation. We have shown this improvement in a Big Data application scenario of the Web user feedback prediction (corresponding to the task typically realized within an RTB DSP). In particular, we have shown that, in the investigated scenario, one may improve the quality of CTR estimation by using a simple, order-three MTHN-based models combining two temporal features sets – the set consisted of the one-hour long nycthemeron (‘day and night’) time slots (i.e., ‘hours’) and the set of week-days (‘days of the week’) – with the set of domains and, equivalently, with the set of advertisers. The high-performance results of the approach applying the

*MTHN(domain)* variant indicate the potential of context-aware modeling – in this case based on the features that describe the Web publishers.

The improvement beyond the state-of-the-art algorithm based on LR (proposed in [23]) was achieved despite the referenced algorithm already involved the use of a basic (not tensor-based) temporal feature model. Additionally, our result was achieved despite the relative simplicity of the tensor-based model. The simplified form of a tensor-based feature representation model presented in this paper does not provide the properties that are widely-regarded as the key source of the practical value of tensor-based data representations. In particular, being used merely as a training data representation structure, the model itself provides no means for tensor-based feature similarity modeling nor tensor decomposition [11, 13] – the techniques that naturally constitute the area of the future research on advancing the model. Moreover, although the use of a multi-tensor hierarchy (see Sect. 3.2) enables performing a sophisticated tensor data centering (which is known as a crucial for effective multilinear data processing [2]), such an application of the model proposed herein is out of the scope of this paper, as well.

Nevertheless, the progress beyond the quality of the state-of-the-art CTR estimation method that has been presented in Sect. 6 indicates that the proposed tensor-based temporal feature model is likely to be worth incorporation into many RTB CTR estimation systems based on LR.

**Acknowledgments.** This work is supported by the Polish National Science Centre, grant DEC-2011/01/D/ST6/06788.

## References

1. Bottou, L.: Stochastic gradient descent tricks. In: Montavon, G., Orr, G.B., Müller, K.R. (eds.) *Neural Networks: Tricks of the Trade*, 2nd edn, pp. 421–436. Springer, Heidelberg (2012)
2. Bro, R., Smilde, A.K.: Centering and scaling in component analysis. *J. Chemometr.* **17**(1), 16–33 (2003)
3. Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): *The Adaptive Web: Methods and Strategies of Web Personalization*. Springer, Berlin (2007)
4. Chapelle, O., Manavoglu, E., Rosales, R.: Simple and scalable response prediction for display advertising. *ACM Trans. Intell. Syst. Technol.* **5**(4), 61:1–61:34 (2014)
5. Cichocki, A.: Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions. *CoRR* abs/1403.2048 (2014)
6. Ciesielczyk, M., Szwabe, A., Morzy, M., Misiorek, P.: Progressive random indexing: dimensionality reduction preserving local network dependencies. *ACM Trans. Internet Technol.* **17**(2), 20:1–20:21 (2017). <http://doi.acm.org/10.1145/2996185>
7. Fawcett, T.: An introduction to ROC analysis. *Pattern Recogn. Lett.* **27**(8), 861–874 (2006)
8. Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: ranking semantic web data by tensor decomposition. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) *ISWC 2009. LNCS*, vol. 5823, pp. 213–228. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-04930-9\\_14](https://doi.org/10.1007/978-3-642-04930-9_14)

9. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., Candela, J.Q.: Practical lessons from predicting clicks on ads at Facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, ADKDD 2014, NY, USA, pp. 5:1–5:9. ACM, New York (2014)
10. Japkowicz, N., Stefanowski, J.: Big Data Analysis: New Algorithms for a New Society. Studies in Big Data. Springer International Publishing, Heidelberg (2015)
11. Kolda, T.G., Sun, J.: Scalable tensor decompositions for multi-aspect data mining. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, ICDM 2008, pp. 363–372. IEEE Computer Society, Washington, DC (2008)
12. Kruczyk, M., Baltzer, N., Mieczkowski, J., Draminski, M., Koronacki, J., Komorowski, J.: Random reducts: a Monte Carlo rough set-based method for feature selection in large datasets. *Fundam. Inform.* **127**(1–4), 273–288 (2013)
13. Lathauwer, L.D., Moor, B.D., Vandewalle, J.: A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **21**, 1253–1278 (2000)
14. Nickel, M., Tresp, V.: An analysis of tensor models for learning on structured data. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23–27, 2013, Proceedings, Part II, pp. 272–287. Springer, Heidelberg (2013)
15. Provost, F., Fawcett, T.: Data science and its relationship to big data and data-driven decision making. *Big Data* **1**(1), 51–59 (2013)
16. Shan, L., Lin, L., Sun, C., Wang, X.: Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization. *Electron. Commer. Res. Appl.* **16**, 30–42 (2016)
17. Shani, G., Gunawardana, A.: Evaluating recommendation systems. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) Recommender Systems Handbook, pp. 257–297. Springer US, Heidelberg (2011)
18. Sutskever, I., Tenenbaum, J.B., Salakhutdinov, R.R.: Modelling relational data using Bayesian clustered tensor factorization. In: Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22, pp. 1821–1828. Curran Associates, Inc. (2009)
19. Szwabe, A., Ciesielczyk, M., Misiorek, P.: Long-tail recommendation based on reflective indexing. In: Wang, D., Reynolds, M. (eds.) AI 2011. LNCS (LNAI), vol. 7106, pp. 142–151. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-25832-9\\_15](https://doi.org/10.1007/978-3-642-25832-9_15)
20. Szwabe, A., Misiorek, P., Walkowiak, P.: Tensor-based relational learning for ontology matching. In: Advances in Knowledge-Based and Intelligent Information and Engineering Systems - 16th Annual KES Conference, San Sebastian, Spain, 10–12 September 2012, pp. 509–518 (2012)
21. Zhang, W., Du, T., Wang, J.: Deep learning over multi-field categorical data. In: Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Nunzio, G.M., Hauff, C., Silvello, G. (eds.) ECIR 2016. LNCS, vol. 9626, pp. 45–57. Springer, Cham (2016). doi:[10.1007/978-3-319-30671-1\\_4](https://doi.org/10.1007/978-3-319-30671-1_4)
22. Zhang, W., Yuan, S., Wang, J.: Optimal real-time bidding for display advertising. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 1077–1086. ACM, New York (2014)
23. Zhang, W., Yuan, S., Wang, J.: Real-time bidding benchmarking with iPinYou dataset. *CoRR abs/1407.7*, pp. 1–10 (2014)

Beyond Databases, Architectures and Structures.  
Towards Efficient Solutions for Data Analysis and  
Knowledge Representation

13th International Conference, BDAS 2017, Ustroń,  
Poland, May 30 - June 2, 2017, Proceedings

KOZIELSKI, S.; Mrozek, D.; Kasprowski, P.;

Małysiak-Mrozek, B.; Kostrzewa, D. (Eds.)

2017, XVIII, 578 p. 184 illus., Softcover

ISBN: 978-3-319-58273-3