Proceedings of the 2014 Federated Conference on Computer Science and Information Systems pp. 129–134
DOI: 10.15439/2014F488
ACSIS, Vol. 2

Dynamic Weighting

New method of weighting panels with large numbers of weighting parameters

Marcin Pery
Military University of Technology
Faculty of Cybernetics
Warsaw, Poland
marcin@pery.pl

Abstract - The algorithm for dynamic weighing presented in this paper is a method used in research studies based on samples when due to the large number of weighting parameters it is not possible to establish a fixed set of sample weights without non-acceptable dispersion of weights.

Keywords: weighting methods, internet audience research, dynamic weighting, machine learning, multiple classifier systems

I. Introduction

Research studies for traditional media have a solid theoretical basis and well-established research methods [1], [2] based on the construction of samples and estimating the number of subsets of the population with specific characteristics. The first step is creating samples - panels [3], [4]. The second step is using known methods of solving systems of equations with many unknowns, determining weights of every element of sample (panelist) in order to make sample "representative" for the entire population [5], [6]. The panel is considered to have better quality when the dispersion of weights is small [7], [8], [9]. The estimation of the number of individuals in the population with the desired attributes (e.g. reading a press title or watching a TV channel) is the sum of the weights of the panelists having these attributes. Internet research studies have their unique characteristics resulting from the number and the type of data, which do not have researchers of other media (e.g. through site-centric type research studies) [10]. Each user's contact with the internet service leaves a trace in servers sending content to the user. Therefore, in the case of internet research studies, we have a vast number of hard data from the measurement of a large number of websites [11]. In practice, the estimation results for the samples (constructed the same way as for the other media) turned out to be unacceptable due to significant discrepancies between the results estimated by using a panel and hard data obtained from the measurement. If the number of weighting parameters is large it is very difficult to correctly determine the weights for a sample. Systems of equations are then unsolvable, or results are unacceptable because the dispersion of weights is too large.

The dynamic weighting algorithm was created in order to estimate coherent and useful results for the whole population (e.g. for audience research studies) even when it is not possible to determine one fixed set of weights of a sample. The algorithm can be used in machine learning and multiple classifier systems where there is a need to draw conclusions on the basis of samples [12], [13], [14], [15].

The very simple example of standard task for computing weights is as follow:

Let us assume that we have a population P of 100 objects which is a set of individuals with two attributes: $\{"is\ men", "is\ woman"\}$. Each of the individuals has exactly one of these two attributes. Let us assume that there are 40 individuals who have attribute "$is\ men$" and 60 with attribute "$is\ woman$". In a sample we have 4 objects, two of which have the attribute "is men" and the other two have the attribute "is woman". To compute weights of objects in the data sample we have to compute systems of equations with many unknowns: $\begin{cases} w_1 + w_2 \approx 40 \\ w_3 + w_4 \approx 60 \end{cases}$

There are infinitely many solutions to this system of equations. We need to find such a solution where weights of objects in a sample are as similar as it is possible. The simplest solution is: $w_1 = 20, w_2 = 20, w_3 = 30, w_4 = 30$. It means that the first object in the sample represents 20 objects in the population, the third object in the sample represents 30 objects in the population, etc.

Let us assume now that there are more attributes (not only sex but also a place of living, age, etc.) and the number of equations is much bigger. There could be no possibility to compute one set of weights which are similar. In a great number of cases some weights have to be set on zero (or close to zero). It means that the sample has very poor quality and as such is of no use. In such cases there is a need to take another approach to determine weights of objects in the sample - this new approach is presented in this very paper.

II. Model

A. Population

Let us define a population of objects P as follows:

$$P = \{p_1, p_2, \ldots, p_N\} \quad (1)$$

Let us define a set of objects' attributes A as follows:

$$A = \{a_0, a_1, a_2, \ldots, a_M\} \quad (2)$$

978-83-60810-58-3/$25.00

Let us define a function which assigns subsets of attributes from set A to objects from set P :

$$a: P \to 2^A \quad (3)$$

Values of function a are known only for some objects from set P. For most of the objects from set P values of function a are unknown.

Attributes are questions about some features of objects. There are only two possible answers: "yes" or "no". If $a_j \in a(o_i)$ it means that the answer is "yes" and the object o_i has the attribute a_j.

The attribute a_0 is special and it is the question: "Does the object belong to the population?". The attribute a_0 meets the following condition:

$$\forall_{o \in P}\left(a_0 \in a(o)\right) \quad (4)$$

Attributes are organized in the *Attributes tree*.

Let us define a function $parent$ which assigns to attribute a_i direct parent node in the *Attributes tree*:

$$parent: A \to A \quad (5)$$

Let us define a function $children$ which assigns a set of direct children nodes in the *Attributes tree* to attribute a_i :

$$children: A \to 2^A \quad (6)$$

Attributes tree meets the following conditions:

- The attribute a_0 is a root of the *Attributes tree*.
- For every pair of p_i and a_j object p_i has no more than one attribute which is the direct child of a_j:

$$\forall_{\mathrm{p_i \in P}} \forall_{\mathrm{a_j \in A}} \overline{\left(\mathrm{a(p_i) \cap children(a_j)}\right)} \leq 1 \quad (7)$$

If object p_i has attribute a_j then object p_i has attribute which is a direct parent of a_j (it does not apply to $\mathrm{a_0}$ as a root of the *Attributes tree*):

$$\forall_{\mathrm{p_i \in P}} \forall_{\mathrm{a_j \in A \setminus \{a_0\}}} \left(\mathrm{a_j \in a(p_i) \Rightarrow parent(a_j) \in a(p_i)}\right) \quad (8)$$

There is non-empty subset of attributes $A^{universe} \subset A$, $\mathrm{A} \neq \emptyset$ that contains only such attributes that we know how many objects from set P have them. Let us define a function $universe$ which assigns number of objects which have the following attribute to the attribute a_i :

$$univers: A^{universe} \to \langle 0, N \rangle \quad (9)$$

$A^{universe}$ meets the following conditions:

$$a_0 \in A^{universe} \quad (10)$$

$$\mathrm{universe(a_0) = N} \quad (11)$$

Let us define a function p which assigns subsets of objects from set P to attributes from set A:

$$p: A \to 2^P \quad (12)$$

$$\forall_{a_j \in A} \forall_{p_i \in p(a_j)} \left(a_j \in a(p_i)\right) \wedge \forall_{p_i \in P} \forall_{a_j \in a(p_i)} \left(p_i \in p(a_j)\right) \quad (13)$$

B. Sample

There is a non-empty subset $S \subset P$ (called *Sample*) that contains only such objects that we know values of function a for these objects. Let us define such a subset as follows:

$$S = \{s_1, s_2, \ldots, s_n\}, S \neq \emptyset \quad (14)$$

Let us define a function w (called *Weighing function*) which assigns the size of part of population which these objects "represent" (called *Weight*) to objects from set S:

$$\mathrm{w: S \to \langle 1, N \rangle} \quad (15)$$

Weighting function meets the following conditions:

- The sum of *Weights* of every element from set S is N:

$$\sum_{i=1}^{n} w(s_i) = N \quad (16)$$

- The sum of Weights of objects which have some attributes must be equal to values of function $universe$ for this attributes:

$$\forall_{a_j \in A^{universe}} \left(\sum_{s_i \in s(a_j)} w(s_i) = universe\left(a_j\right)\right) \quad (17)$$

Let us define *dispersion* e as a quality measure of *Weighting function* for *Sample*:

$$e = \frac{\sum_{i=1}^{n} \left(w(s_i) - \frac{N}{n}\right)^2}{n} \quad (18)$$

If e is big, the quality of sample data is bad and *Sample* cannot be used as a reliable source of information for the whole population P.

C. Task

Let us define *Question* as a subset $\mathrm{Q \subseteq A}$:

$$Q = \{q_1, q_2, \ldots, q_m\} \quad (19)$$

Having given:

- *Question* Q,
- Attributtes tree,
- function a,

- function p,
- *Sample* S,
- function $universe$,
- *Weighting function* w

find number R which is a size of a subset of objects from set P which have at least one attribute from set Q.

D. The trivial solution

If the model satisfies all the above assumptions, including the assumption (17), the number R is computing as follows:

$$R = \sum_{s_i \in \left(\cup_{j=1}^{m} p(q_j) \cap S\right)} w(s_i) \tag{20}$$

III. PROBLEM

If the number of attributes in set $A^{universe}$ is as big that it is almost impossible to satisfy all the above assumptions, including the assumption (17) with acceptable level of *dispersion* e. The set of equations needed to solve this case has no solution or final *Weights* are so different that it is not possible to draw conclusions on the basis of *Sample*.

To be able to compute any reliable results, the assumption (17) is satisfied only for some subset $\mathrm{A^{universe'}} \subset \mathrm{A^{universe}}$. In the internet research studies number of attributes in set $A^{universe'}$ is even tens of times greater than size of $A^{universe}$. Because of that, there is a need to use a different method of constructing *Weighting function* and computing number R than the trivial solution presented above.

IV. DYNAMIC WEIGHTING

A. Assumptions

The basic property of the algorithm is that the *Weights* are calculated based on the questions Q. Depending on what attributes belong to Q, the *Weighting function* will assign different values to objects from *Sample*. Dynamic *Weighting function* meets the following conditions:

1. Monotonicity:

$$\forall_{a_j \in A} R(Q,\dots) \leq R\left(Q \cup \{a_j\},\dots\right) \tag{21}$$

2. Additivity:

$$\forall_{a_j \in A} R(Q,\dots) = R\left(Q \setminus \{a_j\} \cup children(a_j),\dots\right) \tag{22}$$

3. Completeness:

$$\forall_{a_j \in A^{universe'}} \left(\sum_{s_i \in s(a_j)} w(s_i) = universe\,(a_j)\right) \tag{23}$$

B. Algorithm

Input: *Question* Q, *Attributes tree*, function a, function p, *Sample* S, function $universe$.

Task: determine the *Weighting function* w'.

The algorithm is an iterative algorithm providing the ultimate form of the function w' in the steps going from the bottom of the *Attributes tree* to its root.

Step 1. Calculate initial value of the *Weights* w' using classical methods based on the completeness condition (23).

Step 2. If set Q is empty it ends the algorithm.

Step 3. Determine the attribute a^{parent} from set A which is the parent of all the attributes from set Q with the longest path from the root. If Q contains only one element q_j, then $a^{parent} = q_j$.

Step 4. Create the set Q' including:

- all elements from Q ,
- attribute a^{parent},
- all children of attribute a^{parent} which are parents of any attribute from set Q .

Step 5. Each attribute q_j from a set Q' assigns a temporary *Weighting function* w', as follows:

- if $q_j \in A^{universe'}$:

$$\forall_{p_i \in p(q_j)} w'_j(p_i) = w'(p_i) * \frac{universe(q_j)}{\sum_{p_k \in p(q_j)} w'(p_k)} \tag{24}$$

- otherwise:

$$\forall_{p_i \in p(q_j)} w'_j(p_i) = w'(p_i) \tag{25}$$

Step 6. If the set Q' contains only one element q_j, $it\ w'$ assigns to w'_j and it stops the algorithm.

Step 7. Determine from a set Q' such attribute q' for which the distance from the root is the largest (in the case where there is more than one attribute of the longest path, select any of them by any means).

Step 8. Determine the attribute $\mathrm{a'}$ which is the direct parent of q':

$$\mathrm{a'} = \mathrm{parent(q')} \tag{26}$$

Step 9. Modify the temporary *Weighting function* w' for the attribute $\mathrm{a'}$ as follows:

a. create subsets $s'_{positive}$ i $s'_{negative}$ of *Sample* S as follows:

$$s'_{positive} = \cup_{a_j \in children(a') \cap Q'} p(a_j) \tag{27}$$

$$s'_{negative} = \left(\cup_{a_j \in children(a')} p(a_j)\right) \setminus s'_{positive} \tag{28}$$

b. create temporary *Weighting function* w'' as follows:

$$\forall_{p_i \in s'_{positive}} w''(p_i) = max_{\,a_j \in children(a')\cap Q'}\left(w'_j(p_i)\right) \quad (29)$$

$$\forall_{p_i \in s'_{negative}} w''(p_i) = min_{\,a_j \in children(a')\setminus Q'}\left(w'_j(p_i)\right) \quad (30)$$

c. Normalize the *Weighting function* w'' in the way that the sum of *Weights* w'' is $universe(a')$:

$$\forall_{p_i \in S}\, w''(p_i)^{new} = w''(p_i)^{old} * \frac{universe(a')}{\sum_{p_i \in S} w''(p_i)^{old}} \quad (31)$$

d. modify the temporary *Weighting function* w' for the attribute a' as follows:

$$corr = \frac{\sum_{p_i \in s'_{positive}} w''(p_i)}{\sum_{p_i \in s'_{positive}} w''(p_i) + \sum_{p_i \in s'_{negative}} w''(p_i)} \quad (32)$$

$$\forall_{p_i \in S}\, w'(p_i)^{new} = w''(p_i) + corr * (w'(p_i)^{old} - w''(p_i)) \quad (33)$$

Step 10. Delete from set Q' all children of node a' :

$$Q'^{new} = Q'^{old} \setminus children(a') \quad (34)$$

and back to step 6.

C. Solution

Having determined Weighting function w′for a given *Question* Q it is possible to answer the *Question* Q using analogous calculation as in the case of the classical selection of weights:

$$R = \sum_{s_i \in \left(\cup_{j=1}^{m} p(q_j) \cap s\right)} w'(s_i) \quad (35)$$

D. Known features of the algorithm

Where: *Sample* S is selected from the population P using the random function with known distribution, we know the size of the entire population N and the sample size M and the weight of the sample are selected by the classical method satisfying the assumption (17), we are able to calculate statistical errors of estimates of the function R. In the case of Dynamic weighting calculating statistical errors it is difficult due to the complexity and nonlinearity of the algorithm for determining *Weighting function* w'.

Due to the condition of monotonicity (21) the algorithm tends to overestimate the weights of objects that have more than one attribute and whose weights differ in the temporary weights significantly. The better the quality of the *Sample*, the less noticeable the phenomenon is.

E. Example

Input:

Size of population is:

$$\mathrm{N} = 1000 \quad (36)$$

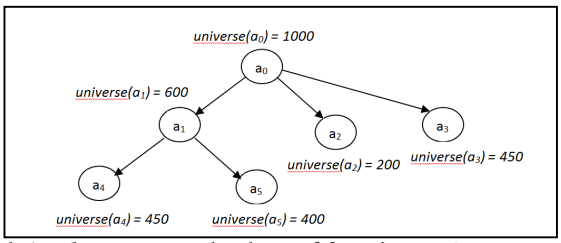
Set of attributes is:

$$\mathrm{A} = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \mathrm{a}_4, \mathrm{a}_5\} \quad (37)$$

where:

$$\mathrm{A}^{\mathrm{universe}} = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3, \mathrm{a}_4, \mathrm{a}_5\} \quad (38)$$

$$\mathrm{A}^{\mathrm{universe}'} = \{\mathrm{a}_0\} \quad (39)$$



and *Attributes tree* and values of function $universe$ are:

Fig. 1. Example: Atrributes tree and values of function universe

Sample is:

$$\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \mathrm{s}_3, \mathrm{s}_4, \mathrm{s}_5, \mathrm{s}_6, \mathrm{s}_7, \mathrm{s}_8\} \quad (40)$$

Values of function a are described by the following table:

TABLE I. EXAMPLE: VALUES OF FUNCTION A FOR SAMPLE OBJECTS

Sample	***Attributes***					
	a_0	**a_1**	**a_2**	**a_3**	**a_4**	**a_5**
s_1	▪	▪			▪	
s_2	▪	▪			▪	
s_3	▪	▪	▪		▪	▪
s_4	▪	▪	▪			▪
s_5	▪		▪			
s_6	▪		▪	▪		
s_7	▪			▪		
s_8	▪			▪		

Question:

How many objects having the attribute a_3 or a_4 are in the population *P*?

$$\mathrm{Q} = \{\mathrm{a}_3, \mathrm{a}_4\} \quad (41)$$

Solution:

Step 1: Based on (32) initial values of the *Weights* w' were calculated as follows:

TABLE II. EXAMPLE: INITIAL VALUES OF WEIGHTS

	w'
s_1	125
s_2	125
s_3	125
s_4	125
s_5	125
s_6	125
s_7	125
s_8	125

Step 3: a^{parent} is determined as follows:

$$\mathrm{a}^{\mathrm{parent}} = \mathrm{a}_0 \tag{42}$$

Step 4: Set Q' was determined as follows:

$$\mathrm{Q}' = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_3, \mathrm{a}_4, \mathrm{a}_5\} \tag{43}$$

Step 5: The temporary *Weighting function* w' was determined as follows:

TABLE III. EXAMPLE: INITIAL TEMPORARY WEIGHTS

Sample	Weights					
	w'_0	w'_1	w'_2	w'_3	w'_4	w'_5
s_1	125	150			150	
s_2	125	150			150	
s_3	125	150	50		150	200
s_4	125	150	50			200
s_5	125		50			
s_6	125		50	150		
s_7	125			150		
s_8	125			150		

Step 7: Attribute a_4 was determined as the node which has the longest path from the root in the *Attributes tree*.

Step 8: Attribute a_1 was determined as the direct parent of a_4:

$$\mathrm{parent}(\mathrm{a}_4) = \mathrm{a}_1 \tag{44}$$

Step 9: The temporary *Weights* w'_1 was modified as follows:

b.

TABLE IV. EXAMPLE: CREATING W''_1

	w''_1	w'_4	w'_5
s_1	150	150	
s_2	150	150	
s_3	150	150	100
s_4	100		100

c.

TABLE V. EXAMPLE: NORMALIZING W''_1

	w''_1
s_1	138
s_2	138
s_3	138
s_4	186

d.

$$\mathrm{corr} = \frac{414}{414 + 185} \cong 0{,}69$$

TABLE VI. EXAMPLE: MODIFYING W'_1

	w'^{old}_1	w'^{new}_1	w''_1
s_1	150	146	138
s_2	150	146	138
s_3	150	146	138
s_4	150	162	186

Step 10: Set Q' was modified as follows:

$$\mathrm{Q}' = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_3\} \tag{45}$$

and went back to step 6.

Step 7^2: Attribute a_4 was determined as the node which has the longest path from the root in the *Attributes tree*.

Step 8^2: Attribute a_0 was determined as the direct parent of a_3:

$$parent(a_3) = a_0 \tag{46}$$

Step 9^2: The *Temporary weights* w'_0 was modified as follows:

b.

TABLE VII. EXAMPLE: CREATING W''_0

	w''_0	w'_1	w'_2	w'_3
s_1	146	146		
s_2	146	146		
s_3	146	146	50	
s_4	50	186	50	
s_5	50		50	
s_6	150		50	150
s_7	150			150
s_8	150			150

c.

TABLE VIII. EXAMPLE: NORMALIZING W''_0

	w''_0
s_1	148
s_2	148

	w''_0
s_3	148
s_4	51
s_5	51
s_6	152
s_7	152
s_8	152

d.

$$\text{corr} = \frac{899}{899+101} \cong 0{,}90$$

TABLE IX. EXAMPLE: MODIFYING w'_0

	w'^{old}_0	w'^{new}_0	w''_0
s_1	125	127	148
s_2	125	127	148
s_3	125	127	148
s_4	125	117	51
s_5	125	117	51
s_6	125	128	152
s_7	125	128	152
s_8	125	128	152

Step 10^2: Set Q' was modified as follows:

$$Q' = \{a_0\} \tag{47}$$

and went back to step 6.

Step 6: Since there is only one element in set Q' T*emporary weights* w'_0 are the final weights for *Sample*:

TABLE X. EXAMPLE: FINAL WEIGHTS FOR SAMPLE

	w'
s_1	127
s_2	127
s_3	127
s_4	117
s_5	117
s_6	128
s_7	128
s_8	128

Answer:

There are 765 objects in population P which have attributes a_3 or a_4.

$$R \cong 765 \tag{48}$$

V. CONCLUSIONS

There are cases where it is not possible to determine one constant *Weights* for *Sample* with acceptable *dispersion* e (and not only in internet research studies). In these cases Dynamic weighting algorithm may determine an individual set of *Weights* for each *Question*. The algorithm satisfies conditions (21) and (22) and (23), which makes the results reliable and useful for applied studies.

The presented algorithm is not protected against possible specific cases and possible incorrect input. Its practical implementation must take into account such cases like inconsistency of information ($e.g. universe(a_j) < \sum_{a_k \in children(a_j)} universe(a_k)$) or even contradictory information ($universe(a_j) > universe\left(parent(a_j)\right)$ at different levels of the *Attributes tree*.

The algorithm is presented in the simplest possible form. In the practice of its implementation in many places it can be optimized in terms of speed as well as memory resource consumption (e.g., through the use of temporary variables that store the temporary results) .

In order to simplify the algorithm, the questions Q are created as the sum of sets of attributes (corresponding to the logical operators OR). Practical implementations can also use the intersections (corresponding to the logical operators AND).

The assumptions of the presented algorithm are used by Gemius SA (research company) in the commercial online research studies in Poland, where the size of the population of internet users is several million people, the sample (panel of internet users) counts several thousand panelists and there are thousands of websites, internet services and web applications presented in final results of the audience research study.

In further work on the algorithm there seems to be a promising direction for estimating the statistical error of the results.

REFERENCES

[1] A. Stuart, Basic Ideas of Scientific Sampling, Hafner Publishing Company, New York, 1962.

[2] L. Kish, Survey Sampling, New York, 1965.

[3] K.W. Brown, P.C . Cozby, D.W. Kee, P.E. Worden, Research Methods in Human Development, Mountain View, CA , 1999.

[4] H. Lohr, Sampling: Design and Analysis, Duxbury, 1999

[5] W.G. Cochran, Sampling Techniques, New York, 1977.

[6] G. Kalton, Introduction to Survey Sampling. Sage Publications Series, No. 35, 1983.

[7] R. Lehtonen, E. J. Pahkinen, Practical Methods for Design and Analysis of Complex Surveys. New York, 1995.

[8] S. Levy, S. Lemeshow, Sampling of Populations: Methods and Applications, New York, 1999.

[9] S. Lohr, Sampling: Design and Analysis. Pacific Grove, 1999.

[10] S. Coffey, Internet audience measurement: a practitioner's view, Journal of Interactive Advertising, 2013

[11] P. Ejdys, T. Cisek, C. Modzelewski, Real Profilee, a new aproach to online media planning, Worldwide Audience Measurement 2003 - Online and Out-of-Home / Ambient Media, 2003

[12] Michal Wozniak, Manuel Graña, Emilio Corchado: A survey of multiple classifier systems as hybrid systems. Information Fusion 16: 3-17 (2014)

[13] Michal Wozniak, Bartosz Krawczyk: Combined classifier based on feature space partitioning. Applied Mathematics and Computer Science 22(4): 855-866 (2012)

[14] Bartosz Krawczyk, Gerald Schaefer: A hybrid classifier committee for analysing asymmetry features in breast thermograms. Appl. Soft Comput. 20: 112-118 (2014)

[15] Konrad Jackowski: Multiple Classifier System with Radial Basis Weight Function. HAIS (1) 2010: 540-547

[16] Dymitr Ruta, Bogdan Gabrys: Classifier selection for majority voting. Information Fusion 6(1): 63-81 (2005)