# Joint Cutoff Probabilistic Estimation using Simulation: A Mailing Campaign Application*

Antonio Bella, Cèsar Ferri, José Hernández-Orallo, and María José Ramírez-Quintana

Universitat Politècnica de València, DSIC, Valencia, Spain

**Abstract.** Frequently, organisations have to face complex situations where decision making is difficult. In these scenarios, several related decisions must be made at a time, which are also bounded by constraints (e.g. inventory/stock limitations, costs, limited resources, time schedules, etc). In this paper, we present a new method to make a good global decision when we have such a complex environment with several local interwoven data mining models. In these situations, the best local cutoff for each model is not usually the best cutoff in global terms. We use simulation with Petri nets to obtain better cutoffs for the data mining models. We apply our approach to a frequent problem in customer relationship management (CRM), more specifically, a direct-marketing campaign design where several alternative products have to be offered to the same house list of customers and with usual inventory limitations. We experimentally compare two different methods to obtain the cutoff for the models (one based on merging the prospective customer lists and using the local cutoffs, and the other based on simulation), illustrating that methods which use simulation to adjust model cutoff obtain better results than a more classical analytical method.

## 1 Introduction

Data mining is becoming more and more useful and popular for decision making. Single decisions can be assisted by data mining models, which are previously learned from data. Data records previous decisions proved good or bad either by an expert or with time. This is the general picture for predictive data mining. The effort (both in research and industry) is then focussed on obtaining the best possible model given the data and the target task. In the end, if the model is accurate, the decisions based on the model will be accurate as well.

However, in real situations, organisations and individuals must make several decisions for several given problems. Frequently, these decisions/problems are interwoven with the rest, have to be made in a short period of time, and are accompanied with a series of constraints which are also just an estimation of the

* This work has been partially supported by the EU (FEDER) and the Spanish MEC under grant TIN 2007-68093-C02-02, Generalitat Valenciana under grant GV06/301, UPV under grant TAMAT and the Spanish project "Agreement Technologies" (Consolider Ingenio CSD2007-00022)

real constraints. In this typical scenario, making the best local decision for every problem does not give the best global result. This is well-known in engineering and decision making, but only recently acknowledged in data mining. Examples can be found everywhere: we cannot assign the best surgeon to each operation in a hospital, we cannot keep a fruit cargo until their optimal consumption point altogether, we cannot assign the best delivering date for each supplier, or we cannot use the best players for three matches in the same week.

In this context, some recent works have tried to find optimal global solutions where the local solutions given by local models are not good. These works address specific situations: rank aggregation [3] and cost-sensitive learning are examples of this, a more general "utility-based data mining"[1] also addresses this issue, but also some other new data mining tasks, such as quantification [5], are in this line. Data mining applied to CRM (Customer-Relationship Management) [1] is also one of the areas where several efforts have also been done.

Although all these approaches can be of great help in specific situations, most of the scenarios we face in real data mining applications do not fit many of the assumptions or settings of these previous works. In fact, many real scenarios are so complex that the "optimal" decision cannot be found analytically. Approximate, heuristic or simplified global models must be used instead. One appropriate non-analytic way to find good solutions to complex problems where many decisions have to be made is through simulation.

In this work, we connect inputs and outputs of several data mining models and simulate the global outcome under different possibilities. Through the power of repeating simulations after simulations, we can gauge a global cutoff point in order to make better decisions for the global profit. It is important to highlight that this approach does not need that local models take the constraints into account during training (i.e. models can be trained and tested as usual). Additionally, we can use data which has been gathered independently for training each model. The only (mild) condition is that model predictions must be accompanied by probabilities (see e.g. [4]) or certainty values, something that almost any family of data mining algorithms can provide. Finally, probabilities and constraints will be used at the simulation stage for estimating the cutoff.

In order to do this, we use the basic Petri Nets formalism [6], with additional data structures, as a simple (but powerful) simulation framework and we use probabilistic estimation trees (classical decision trees accompanied with probabilities [4]). We illustrate this with a very frequent problem in CRM: we apply our approach to a direct-marketing campaign design where several alternative products have to be offered to the same house list of customers. The scenario is accompanied, as usual, by inventory/stock limitations. Even though this problem seems simple at the first sight, there is no simple good analytic solution for it. In fact, we will see that a reasonable analytic approach to set different cutoffs for each product leads to suboptimal overall profits. In contrast, using a joint cutoff probabilistic estimation, which can be obtained through simulation, we get better results.

[1] (http://storm.cis.fordham.edu/~gweiss/ubdm-kdd05.html)

The paper is organised as follows. Section 2 sets the problem framework, some notation and illustrates the analytical (classical) approach. Section 3 addresses the problem with more than one product and presents two methods to solve it. Section 4 includes some experiments with the presented methods. The paper finishes in Section 5 with the conclusions.

## 2 Campaign Design with One Product

Traditionally, data mining has been widely applied to improve the design of mailing campaigns in Customer Relationship Management (CRM). The idea is simple: discover the most promising customers using data mining techniques, and in this way, increase the benefits of a selling campaign.

The process begins by randomly selecting a sample of customers from the company database (house list). Next, all these customers receive an advertisement of the target product. After a reasonable time, a minable view is constructed with all these customers. In this table, every row represents a different customer and the columns contain information about customers; the predictive attribute (the target class) is a Boolean value that informs whether the corresponding customer has purchased or not the target product. Using this view as a training set, a probability estimation model is learned (for instance a probability estimation tree). This model is then used to rank the rest of customers of the database according to the probability of buying the target product. The last step is to select the optimal cutoff that maximises the overall benefits of the campaign, i.e. the best cutoff of the customer list ranked by customer buying probability.

The optimal cutoff can be computed using some additional information about some associated costs: the promotion material cost (edition costs and sending cost)($Icost$), the benefit from selling one product ($b$) and the cost to send an advertisement to a customer ($cost$). Given all this information, the *accumulated expected benefit* for a set of customers is computed as follows. Given a list $C$ of customers, sorted by the expected benefit (for $c_k \in C, E\_benefit(c_k) = b \times p(c_k) - cost$), we calculate the *accumulated expected benefit* as $-Icost + \sum_{k=1}^{j} b \times p(c_k) - cost$, where $p(c_k)$ is the estimated probability that customer $c_k$ buys the product and $j$ is the size of the sample of customers to which a pre-campaign has offered the product. The optimal cutoff is determined by the value $k, 1 \leq k \leq j$ for which the greatest accumulated expected benefit is obtained.

The concordance between the real benefits with respect to the expected benefits is very dependent on the quality of the probability estimations of the model. Therefore, it is extremely important to train models that estimate accurate probabilities (e.g. see [4]). A more reliable estimation of the cutoff can be obtained by employing different datasets of customers (or by spliting the existing dataset): a training dataset for learning the probability estimation models, and a validation dataset to compute the optimal cutoff. With this validation dataset the latter estimation of the *accumulated expected benefit* turns into a real calculation of the *accumulated benefit*, where $p(c_k)$ is changed by $f(c_k)$ in the formula, being $f(c_k)$

the response of $c_k$ wrt. the product, such that $f(c_k) = 0$ if customer $c_k$ does not buy the product and $f(c_k) = 1$ if $c_k$ buys it. Then, the cutoff is determined by the greatest accumulated benefit.

Let us see an example where the benefit for the product is 200 monetary units (m.u.), the sending cost is 20 m.u. and the investment cost is 250 m.u. In Table 1 we compare the results obtained with each method. According to the *accumulated expected benefit* we will set the cutoff at 90% of the customers, which clearly differs from the maximum *accumulated benefit* (located at 70%).

**Table 1.** Accumulated expected benefit vs. Accumulated benefit

| Customer | Buys | Probability | E(Benefit) | Acc. Exp. Benefit | Acc. Benefit |
|---|---|---|---|---|---|
| | | | | -250 | -250 |
| 3 | YES | 0.8098 | 141.96 | -108.04 | -70 |
| 10 | YES | 0.7963 | 139.26 | 31.22 | 110 |
| 8 | YES | 0.6605 | 112.10 | 143.31 | 290 |
| 1 | YES | 0.6299 | 105.98 | 249.30 | 470 |
| 4 | NO | 0.5743 | 94.86 | 344.15 | 450 |
| 6 | NO | 0.5343 | 86.85 | 431.00 | 430 |
| 5 | YES | 0.4497 | 69.94 | 500.94 | **610** |
| 7 | NO | 0.2675 | 33.50 | 534.44 | 590 |
| 9 | NO | 0.2262 | 25.24 | **559.68** | 570 |
| 2 | NO | 0.0786 | -4.29 | 555.39 | 550 |

## 3 Using Simulation and Data Mining for a Campaign Design with More than One Product

The approach shown at the previous section has been successfully applied to very simple cases (i.e. one single product for each campaign), but computing optimal cutoffs by analytic methods is impossible for more complex scenarios (more than one product, constraints for the products, etc.). Therefore, in this section we develop two different approaches: one is an extension of the analytic method, and the other is a more novel and original method based on simulation.

Back on our marketing problem, the objective now is to design a mailing campaign offering $N$ products to a customer list, but taking the following constraints into consideration: there are stock limits (as usual), each product has a different benefit, and the products are alternative, which means that each customer would only buy one of them (or none). As we have seen at Section 2, a good solution, at least apriori, could be to determine a cutoff point defining the segment of customers we have to focus on. But now, since there are several products, it is not clear how this cutoff can be defined/determined. Based on the idea of sorting the customers by their expected benefit, one possibility (what we call the *single approach*) is to combine (in some way, like adding or averaging) the optimal cutoffs which are analytically calculated for each product, in order

to obtain a unique cutoff for the global problem. An alternative method, that we call *joint simulation approach*, is to determine in a dynamic way the global cutoff. We use a validation set to simulate what will happen in a real situation if the customer receives the advertisement (of any of the $N$ products).

Considering that all products have the same sending cost (*cost*), we define the following two alternative ways for obtaining a global cutoff using a validation set $C$:

1. **Single Approach**: For each product $i$, we downwardly sort $C$ by the expected benefit of the customers, obtaining $N$ ordered validation sets $C_i$ (one for each product $i$). Now, for each $C_i$, $1 \leq i \leq N$, we determine its local cutoff point as we have explained in Section 2. Then, the global cutoff $T$ is obtained by averaging the local cutoffs. In order to apply it, we now jointly sort the customers by their expected benefit considering all products at the same time (that is, just one ranked list obtained by merging the sets $C_i$). That produces as a result a single list $SC$ where each customer appears $N$ times. Finally, the cutoff $T$ is applied over $SC$. Then, the real benefit obtained by this method will be the *accumulated benefit* for the segment of customers that will receive the advertisement for the total house list, which will be determined by this cutoff $T$.
2. **Joint Simulation Approach**: Here, from the beginning, we jointly sort the customers downwarded by their expected benefit of all the products, i.e. we merge the $N$ sets $C_i$. However, we do not use local cutoffs to derive the global cutoff, but we calculate the cutoff by simulating $N \times |C|$ *accumulated benefits* considering all the possible cutoffs $T_j, 1 \leq j \leq N \times |C|$, where $T_1$ is the cutoff that only considers the first element of $SC$, $T_2$ is the cutoff that considers the two first elements of $SC$, and so on. Then, the best *accumulated benefit* gives the global cutoff.

To illustrate these two approaches consider a simple example consisting of 10 customers, 2 products ($p_1$ and $p_2$) and the parameters $Icost_{p_1} = 150$, $Icost_{p_2} = 250$, $b_1 = 100$, $b_2 = 200$, and $cost = 20$. Table 2 Left shows for each product the list of customers sorted by its expected benefit as well as the local cutoffs marked as horizontal lines. As we can observe, the cutoffs for products $p_1$ and $p_2$ are 90% and 70% respectively. Table 2 Right shows the global set and the global cutoff, which is marked by an horizontal line, computed by each approach. Note that the cutoff computed by the single and joint simulation methods is different. For the *single approach*, the global cutoff is 80% (the average of 90% and 70%), whereas the cutoff computed by the *joint simulation approach* is 90%.

We have adopted Petri nets [6] as the framework to formalise the simulation. Petri nets are well-known, easy to understand, and flexible. Nonetheless, it is important to highlight that the method we propose can be implemented with any other discrete simulation formalism. We used a unique Petri net to simulate the behaviour of all the customers, but we also implemented additional data structures to maintain information about customers and products (e.g. remaining stock for each product, remaining purchases for each customer). The Petri net

**Table 2.** Left: Customers sorted by their expected benefit for the case of two products. Right: Customers and cutoff for the Single and Joint Simulation Approaches

| **Product $p_1$** | | | |
|---|---|---|---|
| **Customer** | **E(Benefit)** | $f_{p_1}$ | **Acc. Benefit** |
| | | | -150 |
| 2 | 76.61 | 1 | -70 |
| 8 | 75.71 | 1 | 10 |
| 9 | 60.37 | 0 | -10 |
| 5 | 48.19 | 1 | 70 |
| 1 | 44.96 | 1 | 150 |
| 7 | 30.96 | 0 | 130 |
| 10 | 24.58 | 1 | 210 |
| 3 | 23.04 | 0 | 190 |
| 6 | 7.81 | 1 | 270 |
| 4 | -4.36 | 0 | 250 |

| **Product $p_2$** | | | |
|---|---|---|---|
| **Customer** | **E(Benefit)** | $f_{p_2}$ | **Acc. Benefit** |
| | | | -250 |
| 3 | 141.96 | 1 | -70 |
| 10 | 139.26 | 1 | 110 |
| 8 | 112.10 | 1 | 290 |
| 1 | 105.98 | 1 | 470 |
| 4 | 94.86 | 0 | 450 |
| 6 | 86.85 | 0 | 430 |
| 5 | 69.94 | 1 | 610 |
| 7 | 33.50 | 0 | 590 |
| 9 | 25.24 | 0 | 570 |
| 2 | -4.29 | 0 | 550 |

| | **Single & Joint Approaches** | | | |
|---|---|---|---|---|
| | **Customer** | **Product** | **Acc. Benefit** | |
| | | | -400 | |
| | 3 | $p_2$ | -220 | |
| | 10 | $p_2$ | -40 | |
| | 8 | $p_2$ | 140 | |
| | 1 | $p_2$ | 320 | |
| | 4 | $p_2$ | 300 | |
| | 6 | $p_2$ | 280 | |
| | 2 | $p_1$ | 360 | |
| | 8 | $p_1$ | 440 | |
| | 5 | $p_2$ | 620 | |
| | 9 | $p_1$ | 600 | |
| | 5 | $p_1$ | 680 | |
| | 1 | $p_1$ | 760 | |
| | 7 | $p_2$ | 740 | |
| | 7 | $p_1$ | 720 | |
| | 9 | $p_2$ | 700 | |
| Single | 10 | $p_1$ | 780 | |
| | 3 | $p_1$ | 760 | |
| | 6 | $p_1$ | 840 | Joint |
| | 2 | $p_2$ | 820 | |
| | 4 | $p_1$ | 800 | |

can work with as many products and customers as we need with no change in the Petri net structure. Other similar problems, as mailing campaigns with non-alternative products, can also be handled without changes. Figure 1 shows our Petri net which has 24 places and 10 transitions. Each customer arrives to the Petri net and, thanks to the additional data structures created, the suitable number of tokens are put in each place to allow for the suitable transitions to be enabled/disabled and fired or not. E.g. if the remaining stock of the product is not zero a place *P12* is updated with as many tokens as the current stock is, and place *P11* is put to zero. The first place enables the transition *T4* that can be fired if the rest of conditions are fulfilled (place *P14* has a token), while the second place disables the transition *T5* that cannot be fired. Only two arcs have a weight not equal to one, the arc with the *product benefit* and the arc with the *sending cost*. The first arc finishes in the place *P1* (*Total gross benefit*) and the second one finishes in the place *P15* (*Total loss*). The total (or net) benefit for each cutoff is calculated subtracting the number of tokens accumulated in the places *P1* and *P15* (that is, *Total gross benefit* - *Total loss*).

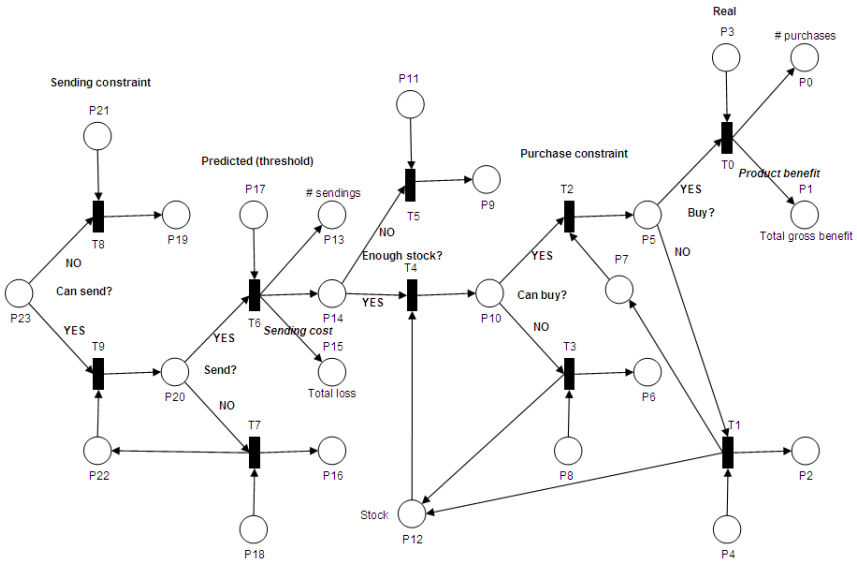


**Fig. 1.** Petri net for our mailing campaign

In this scenario, we consider that, at the most, only one of the $N$ products can be bought since they are alternative products (e.g. several cars or several houses or different brands for the same product). This constraint suggests to offer to each customer only the product with the higher probability of being bought. If we impose this condition then we say that the approach is *with discarding*. In an approach with discarding, only the first appearance of each customer is taken into account. For instance, in the *single approach*, only the first occurrence of each customer in the customer segment determined by the global cutoff is preserved. Analogously, in the *joint simulation approach*, the simulation process does not consider customers that have been already processed. However, since a prospective customer who receives an offer might finally not buy the product, we consider an alternative option which allows several offers to the same customer. This approach is called *without discarding*. The combination of the two approaches and the two options for considering customer repetitions give four scenarios that will be experimentally analysed in the following section. The notation used for referring to these four different methods is: *Single WO* (Single approach without discarding), *Single WI* (Single approach with discarding), *Joint WO* (Joint simulation approach without discarding), and *Joint WI* (Joint simulation approach with discarding).

## 4 Experiments with N products

For the experimental evaluation, we have implemented the four methods explained at Section 3 and the Petri net in Java, and have used machine learning algorithms implemented in the data mining suite WEKA [7].

### 4.1 Experimental settings

For the experiments we have taken a customers file (*newcustomersN.db*) from the SPSS Clementine[2] samples, as a reference. This file has information about only 200 customers, with 8 attributes for each one, 6 of them are nominal and the rest are numeric. The nominal attributes are the *sex* of the customers (male or female), *region* where they live (inner city, rural, town, suburban), whether they are *married*, whether they have *children*, whether they have a *car* and whether they have a *mortgage*. The numeric attributes are the *age* of the customers and their annual *income*.

Since 200 customers are too few for a realistic scenario, we have implemented a random generator of customers. It creates customers keeping the attribute distributions of the example file, i.e. for numeric attributes it generates a random number following a normal distribution with the same mean and deviation as in the example file, and for nominal attributes it generates a random number keeping the original frequency for each value of the attributes in the example file.

Also, to assign a class for each customer (wether s/he buys the product or not), we implemented a model generator. This model generator is based on a random decision tree generator, using the attributes and values randomly to construct the different levels of the tree. We have two parameters which gauge the average depth of the tree and most importantly, the probability of buying each product. We will use these latter parameters in the experiments below.

So, the full process to generate a customer file for our experiments consists of generating the customer data with our random generator of customers and to assign the suitable class with a model obtained by our model generator.

Finally, these are the parameters we will consider and their possible values:

- Number of customers: 10000 (60% training, 20% validation and 20% testing)
- Number of products: 2, 3 and 4
- Probability of buying each product: 0.01, 0.05, 0.2, 0.5, 0.8, 0.95 or 0.99
- Benefits for each product: 100 monetary units (m.u.) for the product 1 and 100, 200, 500 or 1000 m.u. for the other products
- Sending cost (the same for all products): 10, 20, 50 or 90 m.u.
- Stock for each product: 0.1, 0.2, 0.5 or 1 (multiplied by number of customers)
- Investment cost for each product: benefits of the product multiplied by stock of the product and divided by 20
- Correlation (how similar the products are): 0.25, 0.5, 0.75 or 1

### 4.2 Experimental results

The three main experiments consist in testing 100 times the four approaches for 2, 3 and 4 products, where all the parameters are selected randomly for the cases where there are several possible values.

[2] (http://www.spss.com/clementine/)

If we look at overall results, i.e. averaging all the 100 experiments, as shown in Table 3, the results for 2, 3 and 4 products are consistent. As suggested in [2] we calculate a Friedman test and obtain that the four treatments do not have identical effects, so we calculate a post-hoc test (with a probability of 99.5%) This overall difference is clearly significant, as the significant analysis shown in Table 3, illustrates that the joint simulation approaches are better than the single ones. About the differences between with or without discarding methods, in the case of 2 products there are no significant differences. For 3 products the Single WI method wins the Single WO method, and the Joint WO method wins the Joint WI method. In the case of 4 products the approaches with discarding win the approaches without them. Moreover, in the case of 3 products, the Joint WO method clearly outperforms the other 3 methods and, in the case of 4 products is the Joint WI method which wins the rest of methods.

However, it is important to highlight that these values average many different situations and parameters, including some extreme cases where all the methods behave almost equally. This means that in the operating situations which are more frequent in real applications, the difference may be higher than the one reported by these overall results.

Moreover, in the case of 2 products, from the results of the 100 iterations we create three groups taking into account the probability of buying each product (probability of buying the product 1 is greater, equal or less than probability of buying the product 2) and 3 groups taking into account the stocks for the products (stock for the product 1 is greater, equal or less than stock for the product 2). The results obtained are shown in Figure 2. On one hand, the maximum benefit is obtained for all the methods and results are quite similar when the popularity (probability of buying) of both products is the same. On the other hand, the maximum benefit is obtained for all the methods and results are quite similar too when both products have the same stock. The results differ between the four methods especially when probabilities or stocks are different.

**Table 3.** Friedman test: wins (√) /loses (X)/draws(=)

| | **2 products** | | | | **3 products** | | | | **4 products** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **S.WO** | **S.WI** | **J.WO** | **J.WI** | **S.WO** | **S.WI** | **J.WO** | **J.WI** | **S.WO** | **S.WI** | **J.WO** | **J.WI** |
| **Benefits** | 165626 | 164568 | 171225 | 169485 | 182444 | 184077 | 186205 | 185694 | 220264 | 228483 | 231771 | 233724 |
| **S.WO** | - | = | √ | √ | - | √ | √ | √ | - | √ | √ | √ |
| **S.WI** | = | - | = | √ | X | - | √ | = | X | - | = | √ |
| **J.WO** | X | = | - | = | X | X | - | X | X | = | - | √ |
| **J.WI** | X | X | = | - | X | = | √ | - | X | X | X | - |

## 5 Conclusion

In this paper, we have presented a new framework to address decision making problems where several data mining models have to be applied under several constraints and taking their mutual influence into account. The method is based on the conjunction of simulation with data mining models, and the adjustment of model cutoffs as a result of the simulation with a validation dataset. We have

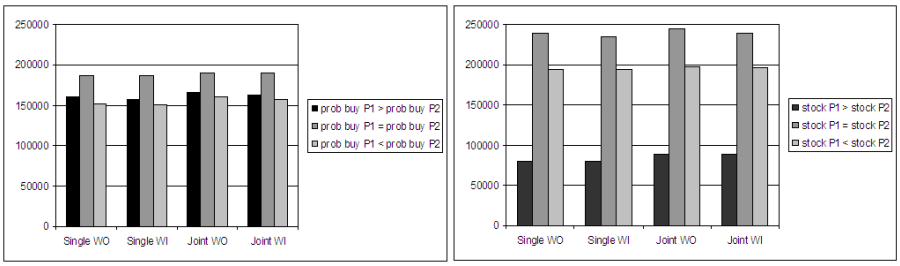


**Fig. 2.** Left: Variations in probability of buying. Right: Variations in stocks

applied this framework to a direct marketing problem, and we have seen that simulation-based methods are better than classical analytical ones.

This specific direct marketing problem is just an example where our framework can be used. Almost any variation of a mailing campaign design problem could be solved (without stocks, with other constraints, non-alternative products, time delays, joint replies, etc.) in some cases with no changes in the presented Petri net and, in the worst case, by just modifying the Petri net that models the constraints and the relations between models. If not only the cutoff is to be determined but also the optimal stock or other important variables, then other techniques, such as evolutionary computation might be used to avoid a combinatorial explosion of the simulation cases. In our example, though, the combinations are not so huge to allow for an exhaustive analysis of all of them.

Out from marketing, we see prospective applicability in many other domains. In particular, the ideas presented here were originated after a real problem we addressed recently in colaboration with a hospital, where resources and data mining models from different services were highly interwoven. Other domains which we are particular familiar with and we plan to use these ideas are the academic world (e.g. university), where we are using data mining models to predict the number of registered students per course each year, but until now we were not able to model the interdependencies between several courses.

## References

1. M. Berry and G. Linoff. *Mastering Data Mining: The Art and Science of Customer Relationship Management*. John Wiley & Sons, Inc., 1999.
2. J. Demsar. Statistical comparisons of classifiers over multiple data sets. *JMLR*, 7:1–30, January 2006.
3. R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *PODS '04: Proceedings of the 32nd symp. on Principles of database systems*, pages 47–58. ACM Press, 2004.
4. C. Ferri, P. Flach, and J. Hernández. Improving the AUC of Probabilistic Estimation Trees. In *Proc. of the 14th European Conf. on Machine Learning*, volume 2837 of *Lecture Notes in Computer Science*, pages 121–132, 2003.
5. G. Forman. Counting positives accurately despite inaccurate classification. In *ECML*, volume 3720 of *LNCS*, pages 564–575. Springer, 2005.
6. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
7. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Elsevier, 2005.

# 6 APPENDIX A. Reviewers Comments

First of all, we want to thank the reviewers for their comments. We explain next the way in which we have implemented their suggestions.

SUGGESTION FROM REVIEWER 2: "However, the authors should give better justification on the usage of Petri nets in the problem in terms of generality". REPLY: We have added some new comments in page 5 on our choice for Petri nets. In fact, this new text explains that Petri nets are just one of the possible choices, namely that "the method we propose can be implemented with any other discrete simulation formalism". We just chose Petri nets because they are "well-known, easy to understand, and flexible".

SUGGESTION FROM REVIEWER 2: "How can we design the Petri nets that are general enough to be used in realistic problems?". REPLY: Page 6 includes revised text on this: "The Petri net can work with as many products and customers as we need with no change on the Petri net structure. Other similar problems, as mailing campaigns with non-alternative products, can also be handled without changes". In the conclusions, we get back on this issue: "Almost any variation of a mailing campaign design problem could be solved (without stocks, with other constraints, non-alternative products, time delays, joint replies, etc.) in some cases with no changes in the presented Petri net and, in the worst case, by just modifying the Petri net that models the constraints and the relations between models.". As we mention in other parts of the paper, Petri nets have been used in an infinite range of domains, so virtually any problem might be modelled using Petri nets, after an appropriate modelling stage.

SUGGESTION FROM REVIEWER 2: "Also, the authors should justify the optimality of the proposed method as they argued in the paper". REPLY: What we argue in the paper is that the experiments show that the joint approach (with simulation) gets better results than the single approach (analytical approach without simulation). The quality of the result depends on the quality of the trained models as any other decision making problem which is based on data mining / machine learning techniques, so the term optimality must be understood in terms of the possible rankings and the order of sending offers that we might organise. In any case, we have revised the paper in order to remove expressions which might cause confusion, such as "the optimal solution" by "a better solution".

SUGGESTION FROM REVIEWER 2: "The experimental part should be also enhanced significantly". REPLY: There are several ways in which the section which describes the experiments can be improved. We have chosen those improvements which fit in a 10-page paper as this. First of all, we have given more support on the claim (also suggested by the reviewer in the comment above) that the joint approach is better. We have included results for 3 and 4 products, which are consistent with the results and claims made in the original submission for 2 products. Additionally, we have changed the traditional student t-tests by more sophisticated (but more rigorous as well) Friedman tests, as suggested by the work from Demsar, which has been recently constituted as the reference on statistical comparison for multiple methods and datasets in the machine learning community. Apart from this, we have rewritten an important part of the discussion on the results to make the experimental results clearer and easy to be interpreted.