Pot-luck challenge: Fact Sheet for the PROMO Dataset

Repository URL: `http://www.causality.inf.ethz.ch/repository.php?id=2`

Dataset name: PROMO

Title: Detecting simple causal effects in time series
Authors: Causality workbench team
Contact: Jean-Philippe Pellet, `jep@zurich.ibm.com`
Website: `http://www.zurich.ibm.com/~jep/causality/promo.html`

Key facts

This dataset contains artificial data about product sales and promotions as time series. There are 1000 binary promotions variables and 100 continuous product sales variables. The goal is to predict a 1000×100 boolean influence matrix, indicating for each (i, j) entry whether the ith promotion has a causal influence of the sales of the jth product.

Abstract

The PROMO dataset proposes the task to identify which promotions affect sales. Artificial data about 1000 promotion variables and 100 product sales is provided. The goal is to predict a 1000×100 boolean influence matrix, indicating for each (i, j) element whether the ith promotion has a causal influence of the sales of the jth product. Data is provided as time series, with a daily value for each variable for three years (i.e., 1095 days).

Each of the 100 products has a defined seasonal baseline, repeating over the years. The seasonal effect can vary from almost inexistent to major. On top of this baseline are promotions. Each product is influenced by between 1 and 50 promotions out of the 1000 promotions available. Promotions usually increase the sales with respect to the baseline, but can occasionally reduce them (e.g., when a similar competing product is promoted, that promotion might have a negative effect on the sales of the current product). On top of that are daily variations.

Each of the 1000 promotions can be seasonal or not; i.e., they can have the same pattern from one year to another or be completely different. The average time a promotion stays active or inactive, however, is constant for each promotion.

The weighted normalized influence matrix is provided for result evaluation. It is normalized so that the maximum positive contribution is 1 and the maximum negative contribution is -1, and each nonzero (i, j) entry is weighted by how much promotion i affects product j.

Note that, as this matrix is provided, the participants are trusted to use it for evaluation purposes only, and not to tune potential hyperparameter of their approaches.

Keywords: time series, structural equation models

Data Generation

The data is generated in three steps:

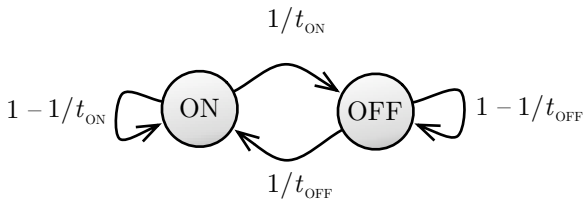

Figure 1: The Markov chain generating the promotion variables

1. Generate the 1000 promotion variables;
2. Generate the product baselines (without the promotion effect);
3. Generate the end product sales, including the promotion effect.

We denote promotion variables by P_i, $1 \leq i \leq 1000$, and the baselines and product sales by B_j and S_j, respectively, $1 \leq j \leq 100$. The value generated for variable P_i on day t is denoted by p_{it}.

The promotion variables are all generated according to a Markov chain whose parameters are randomly chosen. The Markov chain has two states, ON and OFF. The two transition probabilities are determined by the inverse of the average number of days in each state t_{ON} and t_{OFF}, which are drawn from a probability distribution covering from 1 day to 300 days This fully determines the Markov chain:

$$\begin{aligned} p_{\text{ON}\to\text{OFF}} &= 1/t_{\text{ON}} & \qquad p_{\text{OFF}\to\text{ON}} &= 1/t_{\text{OFF}} \\ p_{\text{ON}\to\text{ON}} &= 1 - 1/t_{\text{ON}} & \qquad p_{\text{OFF}\to\text{OFF}} &= 1 - 1/t_{\text{OFF}}. \end{aligned}$$

Then, for each promotion variable, with probability 0.5, it is set to repeat each year in the same pattern as the previous year, and with probability 0.5, not to repeat automatically. In the former case, a full year (i.e., 365 values, one for each day) is sampled by determining the state of the variable according to the Markov chain, and then replicated twice, to obtain the time series over 1095 days. In the latter case, the full 1095 days are sampled with the Markov chain, resulting with high probability in different sequences for each year. In each case, the initial state is determined to be ON with probability $p_{\text{ON}} = t_{\text{ON}}/(t_{\text{ON}} + t_{\text{OFF}})$, and accordingly OFF with probability $p_{\text{OFF}} = 1 - p_{\text{ON}}$. This is show in Figure 1.

$$\forall i : p_{it} = \text{time series sampled with Markov chain}$$

Product baselines are the sum of a constant factor c_j and of a seasonal effect. The seasonal effect repeats over the years. The baselines indicate what the sales would be, without promotions and without random noise. The constant factor is drawn randomly, and the seasonal effect is determined as a superposition of n sines whose amplitude α_k, phase ϕ_k, and pulse ω_k are drawn randomly. The number of sines n is drawn uniformly between 2 and 10. The seasonal effect is then shifted so that its minimum is 0. This is indicated with the shift$(\cdot)$ function, which we define as $\text{shift}(x_t) = x_t - \min_{t'} x_{t'}$.

$$\forall j : b_{jt} = c_j + \text{shift}\left(\sum_{k=1}^{n} \alpha_k \sin(\omega_k \cdot t/365 - \phi_k)\right)$$

The end sales are generated as follows: for each product, a set $\mathbf{I}_j$ of influencing promotion variables is drawn at random, with its cardinality m uniformly distributed between 1 and 50. The influence f_{jl} of each influencing promotion I_{jl}, $1 \leq l \leq m$, is drawn randomly between 0.2 and 0.8, and negated with probability 0.1. For each day, the total promotion factor τ_{jt} is determined as the square root of the sum of the factors of all influencing promotions whose state is ON. Random Gaussian noise with mean 0 and standard deviation 0.1 is then added to this promotion factor. The end sales are then the product baseline multiplied by the total noisy promotion factor (not that this means that the promotion effect is thus multiplicative rather than additive).

$$\begin{aligned}
\forall j : \mathbf{I}_j &= \text{random set of } m \text{ promotion variables} \\
\forall j, m : f_{jl} &= \text{factor of influence for the } l\text{th promotion in } \mathbf{I}_j \\
\forall j : \tau_{jt} &= \sqrt{\sum_{l=1}^{m} f_{jl} \cdot \mathbf{1}_{p_{\text{ind}_j(l),t}=1}} \\
\forall j : u_{jt} &= t \text{ realizations of a variable } U \sim \mathcal{N}(0,1) \\
\forall j : s_{jt} &= b_{jt} \cdot (\tau_{jt} + u_{jt})
\end{aligned}$$

The value of $\mathbf{1}_{p_{\text{ind}_j(l),t}=1}$ is 1 whenever the lth promotion for product j is ON on day t, and 0 otherwise (the notation $\text{ind}_j(l)$ just converts the product-specific promotion index l for product j to the global, product-independent promotion index).

The final data available to challenge participants are the end sales s_{it} and the promotion variables p_{it}; all other intermediary values remain hidden.

Discussion

There are several ambiguities in the data. For instance, all promotions that repeat year-to-year can be seen as seasonality. Further assumptions are needed here to tell if some observed recurring effect is due to seasonality or to a seasonal promotions. Another problem is that some promotion with a nonzero effect might be ON or OFF all the time, preventing learning algorithms from assessing its effect.

These points are deliberate and correspond to real-life scenarios. Often, products both have a seasonality, and often, the promotions applied to these products in the past also had a certain seasonality. It is therefore important to include an appropriate criterion for to tell these two effects apart. It is also necessary to have an algorithm that can correctly identify promotions whose effect cannot be assessed.

Note that the promotion effect is straighforwardly applied to the end sales: only the current day is used. A given promotion can only have an impact the day it is ON; the sale history has no memory of past promotions. This information was not given to the challenge participants.

Approaches Used by Participants

Two approaches were proposed to solve the PROMO task. They are briefly summarized here; more details can be found on their respective fact sheets at `http://www.clopinet.com/isabelle/Projects/NIPS2008/home.html`.

The first approach, A_1, first tries to extract the baseline by modeling it as an offset-plus-sine for each product, to which is then added the promotion effect:

$$\forall j : b_{jt} = c_j + \alpha_j \sin(\omega_j t + \phi_j)$$
$$\forall j : s_{jt} = b_{jt} + Bu_t,$$

where B is the influence matrix and u_t represents the state of the promotions. This is solved in two steps: first, the parameters $c_j, \alpha_j, \omega_j, \phi_j$ are estimated by fitting the data with the offset-plus-sine model; then, fixing those parameters to the obtained value, B is estimated solving j independent convex problems, subject to a sparsity contraint on B: for each promotion, the number of nonzero entries in B should not be greater than 50 (The number 50 is given in the problem description as upper bound on the number of relevant promotion variables). See Markovsky (2008) for more details as well as the whole source code to reproduce the results listed below.

The second approach, A_2, also consists of two steps, where first the seasonal component is removed, and then the relevant promotion variables are determined. The baseline is modeled as a constant plus a superposition of 16 sines and cosines with different frequencies. Denote a design matrix $Z = [z_1, z_2, \cdots, z_{1095}]^T$, where

$$z_t = \begin{pmatrix} 1 & \sin(2\pi t/365) & \cos(2\pi t/365) & \cdots & \sin(10\pi t/365) & \cos(10\pi t/365) \end{pmatrix}^T,$$

then the baseline is estimated as $\hat{B} = (b_{jt}) = Z(Z^T Z)^{-1} Z^T S$, where $S = (s_{jt})$ is the matrix containing the end sales. The input to the second step of the method is the residuals of this regression, namely $Y = S - B$. The second step selects the relevant promotion variables for each product: this is done with an iterative stepwise selection. The hyperparameters of this selection is then chosen according to an EBIC criterion. See Yin et al. (2008) for more details about this method.

Results

To compare the results of the participants, we used the following metrics: for each of the 100 products, we determine the precision, recall, and F-score of the participants' solution.

The *precision* is a real value between 0 and 1 determining, out of the set of promotion variables proposed by a participant as influencing product j, what proportion of them are actually promotion variables that were in $\mathbf{I}_j$; i.e., which were also used in the generating model to determine the end sales. The precision for product j is then:

$$pr_j = \frac{\text{number of correctly identified promotion variables}}{\text{total number of identified variables}}.$$

The *recall* is a also a real value between 0 and 1 determining how complete the participants' solution were. It is defined similarly as:

$$re_j = \frac{\text{number of correctly identified promotion variables}}{\text{total number of promotion variables used in the generating model}}.$$

	A_1 (Markovsky, 2008)	A_2 (Yin et al., 2008)
Precision	0.38 ± 0.24	0.89 ± 0.14
Recall	0.32 ± 0.23	0.78 ± 0.17
F-score	0.31 ± 0.19	0.82 ± 0.13

Table 1: Mean and standard deviation of the precision, recall, and F-score for the two participants

A perfect solution has precision = recall = 1. A solution with precision = 1, recall = 0.5, for instance, means that all identified promotion variables were indeed correct, but that they only constituted 50% of those actually used in the generating model. Conversely, a solution with precision = 0.5 and recall = 1 is such that although all relevant variables were identified, 50% of all identified variables were not used by the generating model.

Finally, the *F-score* is the harmonic mean of precision and recall:

$$F_j = \frac{2 \cdot pr_j \cdot re_j}{pr_j + re_j}.$$

For the two participants, using approaches A_1 and A_2, the precision, recall, and F-score was evaluated for each product. Table 1 shows the mean and standard deviation of those measures aggregated over all products.

Clearly, A_2 performs much better, getting twice as good both precision and recall. This can be due to a number of reasons: probably, extracting a baseline as a superposition of several sines and cosines rather than a single sine can better recover the original baseline as generated by the model, as the model used a superposition of sines with different amplitudes, phases, and pulses. The residuals obtained after baseline extraction by A_1 still contain a bigger part of the seasonal components than the residuals obtained by A_2. Taking in more promotion variables to try and compensate for a baseline detection that could be better then lowers the precision, while at the same time, not detecting the baseline correctly will tend to lower the recall, as it becomes less likely to be able to make out well the effect of the truly influencing promotion variables.

References

I. Markovsky. Results on the PASCAL challenge "Simple causal effects in time series". Technical report, University of Southampton, 2008.

J. Yin, S. Wang, W. Deng, Y. Hu, and Z. Geng. Iterative stepwise selection and threshold for learning causes in time series. Technical report, Peking University, 2008.