Dynamic Pricing on Online Marketplaces

[Strategies and Applications]

Markus Petrykowski, Nikolai J. Podlesny, Sebastian Rehfeldt, Jan Selke

ABSTRACT

This report elaborates how to use logistic regression to estimate sales probabilities for books based on real data. For this purpose, important features and ways to evaluate the quality of a training model will be assessed. Based on the sales probabilities, the bellman equation will be used to predict the optimal prices.

1. INTRODUCTION

As more and more businesses are being influenced by the digital transformation, vendors have to compete with one another on an even more transparent and embattled platform. This huge amount of competition makes it harder to sell the own goods and forces the participants to keep an eye at each other all the time. Furthermore, it enables the digitalization of the selling process. Therefore, it is crucial to find a way to deal with the huge amount of data automatically.

With techniques of machine learning, in this case logistic regression, it is possible to analyze data in relation to the sales and create a model from which sales probabilities can be derived. Based on this knowledge the optimal price for each market situation according to the created model can be calculated.

The content of this elaboration shall be structured as described in the following. Starting off by introducing the present dataset and its' cleansing in chapter 2, we will illustrate regression based on raw data in chapter 3, followed by describing our approach of improving the feature set in 4 as well as its' evaluation in 5. Further, we will outline the underlying bellman equation used as algorithm in 6 and its' optimization for better performance. Finally, this elaboration will be rounded off by an introduction into our UI in chapter 7 as well as a conclusion in 8.

2. DATASET

The dataset as basis of the model primarily consists of one table that contains real world market situations for certain books. One row corresponds to one market situation for one book and has 81 columns overall.

General	Own	Competitors 1-10
isbn10	sales_rank	offer_01_condition
datum_uhrzeit_von[1]	offer_quality	offer_01_price
datum_uhrzeit_bis[2]	offer_price	offer_01_shipping
		offer_01_is_prime
sold_yn	shipping_time	offer_01_quantity
sold_quality	feedback_count	offer_01_feedback
sold_price	rating	offer_01_rating
offers_total_count		
offers_used_count		

Table 1: Columns of the dataset

A market situation is characterized by some general information, such as the time interval or whether the book was sold. Furthermore, it contains data that describe each offer for this book from ourselves and our competitors including the quality, price, shipping-time and other values shown in table 1. Each of the 3.2 million market situation has a maximum of 10 competitors.

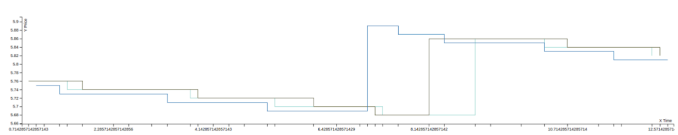

Figure 1: Sample price data for one book with multiple competitors

For each of the books in the present dataset a graph can be drawn as seen in figure 1. It plots the development of the price for each of the competitors over the time. Figure 1 shows a sample where the competitors implemented a pricing strategy each. It has the characterized steps that can often be seen when suppliers try to underbid themselves until they reach a certain threshold and rise up the price again. The selected market situation is one of the few examples in our dataset where other competitors than us follow certain strategies.

2.1 Data Cleansing

In order to create a model that fits the data the best, some data tuples need to be ignored that would otherwise negatively influence the model.

The first criteria to look for at this point is the time period of a certain situation. In general, the shorter the time frame the more unlikely it is to sell. Especially when two market situations are identical, except for the time span, the shorter one would "pollute" the model because it would be treated as if it had an equal chance of triggering a sell. Therefore, each market situation that covers a shorter period than 30 minutes is sorted out. For simplicity reasons other time spans which are in a range between 30 and 120 minutes will be treated equally well. This step reduces the overall number of rows by 31.326 situations.

The next criteria to filter out is the number of existing competitors for a certain market situation. It is important to have some competitor information to be able to evaluate a situation otherwise the ranks would become unreliable. That is why we decided to ignore every state that contains less than 3 competitors which is an overall of 299.801 rows. Another important criteria is the price of the book. We are only interested in books that do create some realistic revenue. Hence, prices of 1 cent or similar, that exist within the dataset, are not relevant for our model. So every row with prices under 75ct will be left out, which deletes another 155.908 elements. By choosing the prices that should be considered one are able to fit the model to certain price ranges. Strategies may differ if dealing with higher priced books than with lower priced ones. But as we want an overall good model we just ignore the cheap priced rows. All in all, around 500.000 rows were removed, but there are still over 2.7 million entries left.

3. REGRESSION BASED ON RAW DATA

Based on the preprocessed dataset, a regression model was built to explain the variance in the sales of the dataset. Therefore, we used the *glm*[3]-function of the programming language R.

Since the time intervals in the dataset are of a rather short duration, the event of realizing a sale is rare. In fact, every observation either has a sale or not but never two or more. Therefore, the outcome variable is binary. This characteristic of the dependent variable in our use case is the reason why the usage of logistic regression is appropriate in our approach. We used the logistic regression of the *glm*-function in R mentioned above with the argument *family = "binomial"*.

3.1 Discovery of first features

Since the model shall predict the probability of selling a book in a certain market situation, the regression model has to explain the variance in the sold_yn variable of the given dataset. As a first approach, the already existing columns were used as explaining variables including Amazon's sales rank, which is a measure assigned by Amazon representing how popular a book is compared to all other books. Additionally, the price and quality information was included for the offered book as well as the total count of offers within the competition.

[3]https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html

Deciding manually for each explaining variable whether it should be included or not represents a huge effort. Instead, the correlation of all attributes with the sold_yn variable shown in table 2 was computed. In this way, one is easily enabled to quantify and estimating the relevance of those variables to realize a sale.

Variable	Correlation with sold_yn
sales_rank	-0.018
offer_counts	0.01
offer_price	-0.009
offer_quality	-0.009

Table 2: Correlation of attributes with sold_yn

Moreover, we assumed that the effect of the sales_rank on the sale is not entirely linear. We believe that the increase in sales probability is stronger when the sales_rank is low and diminishes as the sales_rank rises higher. To reflect this assumption, the logarithm of the sales_rank was incorporated into our regression model.

3.2 Evaluation of the regression models

Apart from checking the correlation, additional means to evaluate the quality of different models were used. We relied on the Akaike information criterion (AIC) which supported our process of model selection. This AIC-score expresses an estimation of how much information is lost in the investigated model compared to the underlying dynamic that generated the dataset. The following formula represents the calculation of the AIC-score where L is the maximum value of the likelihood function and M is the number of explaining variables:

$$AIC = (-2)log(L) + 2M$$

Using such metric enables us on the one hand to reward a good fit of a model and on the other hand to punish an increased amount of explaining variables. Without this penalty, an increasing amount of explaining variables would in most cases result in a growth of fitting accuracy but simultaneously increases the risk of overfitting. The latter especially applies when the number of explaining variables is greater than the amount of observations in the dataset. Thus, when comparing two models based on the same dataset, the one with the lower AIC-score has the better quality in predicting the sales probability [1].

To evaluate the influence of a single explaining variable on the sales probability, the assigned β-coefficient for the correlating variable of the regression model is already providing an estimation for the magnitude of the impact on the sales probability. However, the reader's attention shall be pointed to the matter, that the influence of a variable cannot always be inferred as particularly small or negligible only because the β-coefficient appears to be small. Instead, the β-coefficient should be assessed together with the scope of the explaining variable. When the variable represents a very large value even a small β-coefficient can have a remarkable effect on the output variable. Abstracting from these exceptions, a general rule of thumb can be that an explaining variable is of lesser importance the closer its β-coefficient is to zero.

3.3 Comparison of different models

To obtain different models various experiments were carried out, each with some of the explaining variables mentioned in 3.1 included or not. The resulting models were applied on the preprocessed dataset. Two of these combinations are attached in Table 3.

Combination	AIC-score
log(sales_rank)	82601
log(sales_rank) + offer_quality + offer_price + offers_used_count	81208

Table 3: Combinations of different explaining variables from raw data and respective AIC-score

4. QUALITY IMPROVEMENT WITH NEW FEATURES

The previous chapter demonstrated how it is possible to predict the sales probability using only the raw data. However, the underlying regression model did not take advantage of all the available information in the dataset. In order to improve the quality of the prediction for the sales probability, we aimed to reflect all of the available information in the explaining variables of our regression model by adding new features.

4.1 Introduction of new features

When setting the price of our offered book into relation with the prices of the competitors, a first approach is to calculate a price_rank variable reflecting our offer_price as position within the competition. Similar to the sales_rank described in 3.1, a non-linear correlation between the price_rank and the sales probability was assumed. One may notice that the increase in sales probability is significantly higher when the offer_price is on the first price_rank position or among the top, whereas the positive effect on the sales probability diminishes when the price_rank rises higher and the offers of the competition become cheaper. Therefore, the logarithm of the price_rank as explaining variable was included.
To represent the effect of the overall price level within the competition for an offered book, we introduced the median of all competitor prices as explaining variable. Additionally, two variables for the difference of the offer_price to the smallest competitor price and the difference to the median of all competitor prices were added. In order to capture the price density of a certain market situation in our regression model, we calculated the standard deviation of all competitor prices. Another explaining variable signifies whether the offer_price ends with a 9 as last digit. This variable shall be called psychological_price and we assume that it has a positive effect on the sales probability.
All of the aforementioned new features are based on the price of our own offer and the competitors, but the dataset also provides additional information about other dimensions like the quality of a book, its shipping time or the feedback and rating of a vendor. For capturing this information of the dataset in the regression model, a new explaining variable was created that relates the quality of our offered book with the quality of the competing offers and thereby stating the quality_rank of our offer. Additionally, the same was done with the shipping time information by creating a shipping_rank that expresses how fast our shipping time is compared to the competition. To incorporate the feedback information, one may also create a variable that ranks our own feedback count with those of the competing vendors. To reflect the rating information, a ranking variable was introduced representing how good our own rating is compared to the competition. Since the dataset did not contain information about our own feedback count and rating, we assumed a feedback count of 60000 and a rating of 98-100% in consultation with the data owner.
It should be noticed that all of these four ranking variables were implemented in a way that whenever the value of our own offer has the same rank as a competitor, our own rank will be decreased to the maximum of all competitors who share our rank. This should be a valid extension since - in case a competitor scores the same rank as our own offer - the sales probability will decrease as the former rank position is less distinguishing from the competition and thereby diminishes our advantage.

4.2 Implementation

Unlike features already present in the dataset, all of the above mentioned new features had to be first computed before being used as explaining variables. Due to the large size of the dataset that means a high computational effort. In fact, the computation for the new features took several hours in the first implementation approach when a single thread and data-frames were used. To reduce this effort, the dataset was split up into chunks and parallelization was applied. Together with a change from data-frames to matrices as data structure, this led to a significant speedup regarding the computation of the new features. The calculation after this optimization steps takes about 15-20 minutes, depending on the number of cores used for the computation.

4.3 Comparison of combinations

As next step different combinations of explaining variables were probed. In addition to the already available features of the dataset new features were computed and combined as mentioned in 4.1. Again these have been applied to the preprocessed dataset. Table 4 shows some combinations and their respective AIC-scores.

Combination	AIC-score
previous combination (Table 3)	81208
median+diff_to_min+diff_to_median+ price_density+price_rank+log(price_rank)+ sales_rank+psychological_price	77657
previous+log(sales_rank)+quality_rank+ log(quality_rank)+feedback_rank+ log(feedback_rank)+rating_rank+ log(rating_rank)+shipment_rank+ log(shipment_rank)+offer_quality+ offers_total_count+offers_used_count	76219

Table 4: Different explaining variables combined with new features and respective AIC-score

4.4 Choosing the right explanatory variables

One might think from watching the combinations in the section above that adding more and more new features as explaining variables will always improve the quality of the model and lower the AIC-score. This is, however, not true. Like already mentioned in 3.2, the metric of the AIC-score punishes an increased amount of explaining variables. Also, in order to have stable and reliable results from the logistic regression, the different explaining variables should have little or no correlation with each other. This prohibits to include new features that are too similar to already existing ones without bringing any new information into the model. Furthermore, the computation of the regression model in the R programming language took serious performance hits when we experimented with larger amounts of explaining variables. In case of too many variables were included, the increased calculation time represented an obstacle towards an iterative exploration of explaining variables and their impact. Additionally, the limited main memory capacity of our hardware also restricted the number of features that could be included into the regression model. This lead us to our final combination of explaining variables:

median+diff_to_min+diff_to_median+price_density+
price_rank+log(price_rank)+sales_rank+log(sales_rank)+
psychological_price+quality_rank+log(quality_rank)+
feedback_rank+log(feedback_rank)+rating_rank+
log(rating_rank)+shipment_rank+log(shipment_rank)+
offer_quality+offers_total_count+offers_used_count

5. MODEL EVALUATION

A good model is essential to calculate realistic sales probabilities and to estimate the optimal prices. This section evaluates and discusses the different models presented in the previous sections and introduces a new evaluation method.

5.1 Limits of AIC and beta evaluation

In the former evaluations, the regression coefficients and the AIC-score were used to determine the quality of a specific model, but these criteria have their limits. The β-coefficient can only help by determining the impact of a variable inside a model, but they cannot be used to compare different models against each other. To overcome this problem, the AIC-score was used. Unfortunately, the AIC-score depends on the number of tuples in the dataset and is lower if the dataset contains less tuples. Hence, the AIC-score of the model based on the pre-processed dataset is much smaller than the one from the raw dataset, but we cannot draw conclusions whether the pre-processing leads to a better model.

5.2 Pseudo R^2-value

In linear regression models the *coefficient of determination*, denoted as R^2, is used to indicate how much variance of the dependent variable is explained by the model.[4] The R^2 value is a value between 0 and 1 where a higher value corresponds to a better model. In the case of logistic regression, no equivalent to the R^2-value exists, but it is possible to estimate more general regression models with the help of maximum-likelihood estimation for which *pseudo coefficients of determination*, pseudo-R^2, are proposed[5]. The pseudo-R^2-values are also between 0 and 1 and therefore applies the higher the value, the merrier the model fits the data. The score denotes the improvement in the explained variance from the null-model to the regression-model. To evaluate the models in this work, the McFadden pseudo-R^2-value is used and calculated with the pR2-function from the *pscl* R package[6]. The score is calculated given the following formula[7]:

$$R^2_{mcFadden} = 1 - \frac{LN_1}{LN_0},$$

where LN_1 is the log-likelihood-value of the regression model and LN_0 is the log-likelihood of the null model. The null-model assumes that the explanatory variables are completely independent from the variable we want to explain, whereas the regression-model assumes the opposite. The log-likelihood is lower when the model fits the data better and from the formula you can derive that the score is higher when the regression model fits the data better than the null-model.
To interpret the score, it is necessary to know that a value from 0.2 to 0.4 already indicates a very good fit of the model.

5.3 Discussion of the different models

This section compares the model described in 3.3 with the final model introduced in 4.4 and evaluates the impact of pre-processing on the quality of the model. The pseudo-R^2-values are given in table 5.

Model	no pre-processing	with pre-processing
first model	0.058	0.063
final model	0.116	0.121

Table 5: Pseudo-R^2-values for the different models

The model based on the variables, which are directly available in the dataset, achieved a pseudo-R^2-value of 0.058 on the raw data and 0.062 on the pre-processed data. Also in the final model, the pre-processing leads to a slightly better model. Nevertheless, the pre-processing is quiet useful as it reduces the size of the dataset and leads to faster calculations of the regression model and the additional variables.
The pseudo-R^2-value of the final model is nearly twice the value of the initial model and shows that the new variables led to a much better model, but is still lower than the 0.2 threshold indicating a good fit. This, however, can be explained by the short time intervals in the dataset. Even if the market situation is perfect, it is unlikely to have a sale because of the short time span. Sales are in this way kind of random which makes it really hard to learn a good regression model.

5.4 Clustering according to prices

An idea to further improve the estimation of sales probabilities is to cluster the data into different groups. A model might fit the data rows of cheap books but might have a bad fit on medium- or high-priced books. In future work, it could be analyzed whether a clustering according to price categories or other criteria leads to better models and a more accurate sales prediction.

[4]https://en.wikipedia.org/wiki/Coefficient_of_determination
[5]https://de.wikipedia.org/wiki/Pseudo-Bestimmtheitsmaß
[6]https://cran.r-project.org/web/packages/pscl/pscl.pdf
[7]https://de.wikipedia.org/wiki/Pseudo-Bestimmtheitsmaß#McFadden_R2

6. BELLMAN EQUATION

After creating a model that enables us to calculate sales probabilities for certain market situations, the best price for a given constellation shall be calculated. One possibility to do so is the Bellman equation shown in figure 2.

$$V_t(n,\vec{s}) = \max_{a\geq 0}\left\{\sum_{i=0,1}\underbrace{\pi(i,x(a,\vec{s}))}_{probability}\cdot\left(\underbrace{\min(n,i)\cdot a}_{today's\ profit} - \underbrace{n\cdot l}_{\substack{holding\\costs}} + \underbrace{\delta\cdot V_{t+1}\left(\max(0,n-i),\vec{s}\right)}_{disc.\ exp.future\ profits\ (n-i)^+\ items}\right)\right\}$$

Figure 2: bellman equation

Apart from the sales probability which is calculated using the logit model, it takes three important other things into account. First, the today's profit which is calculated as product of sold items and their prices. Second, the holding costs of the books which are calculated by the amount of offered books times the costs for holding a book for a time interval. In our case, the holding costs for a book are 0.1ct per month. Given the time intervals of 2 hours, one ends up with $l = 0.000002777$€ per book and time interval. As third part, the future profit that derives from the disposal of the other not sold items is added. The discount factor *delta* is in our case 0.99999 and is calculated given the following formula:

$$\delta = \left(\frac{1}{1+r/100}\right)^{1/n},$$

where n is the number of periods per year (=4320) and r is the interest rate per year (=3).
We made use of two different approaches to solve this equation.

6.1 Approximation

The given equation relies on an iterative computation of the optimal prices and is best computed using a dynamic programming approach. With the help of dynamic programming, the value matrix V, consisting of n columns rows (number of books) and t rows (number of iterations), is filled. The last row of the matrix is initialized with a first estimated price $(10 * \sqrt{(column_{id}-1)^2})$. Then, the matrix is filled from the last to the first row and from the first to the last column. Therefore, the values are estimated by the maximum of the revenues generated by all possible prices. Iteratively, the values when offering i items in the matrix converge to the optimal value and hence, the optimal value will be available in the first row. One important assumption to mention is that even if more than one book is offered, we only consider the probabilities to sell none or exactly one book as the sales probability for two or more books is so small that the product in the sum would be negligible.
When the value matrix is filled, it is possible to derive the price which is needed to achieve the optimal value. Therefore, we iterate for an available number of books overall all possible prices, compute the value and check if it matches the optimal value. If we have a match, the current price is used as optimal price. In the end of the approximative approach, we have two arrays. The first contains the optimal revenue which can be achieved depending on the available number of books and the second contains the optimal prices to accomplish these revenues.

6.2 Market situation

The optimal prices and revenues from the previous part are dependent to a specific market situation. A change in the situation would lead to a change in prices and revenues. To quickly react to a new market situation, the calculation of the optimal prices should be fast. We implemented the approximative bellman equation in R and had runtimes of around a minute for 100 iterations and prices from 1 to 20€ in 0.1€ steps even if we used the dynamic programming approach, which is already too slow for us. Considering that we want prices up to over hundred Euro and need around 100 times more iterations to converge, the problem gets really worse. To improve the runtime, we changed each possible for-loop into a *sapply*-function which is faster in R and implemented as much as possible in C++. In the end we needed for 10,000 iterations and prices up to 250€ (0.2€ steps up to 40€ and then 1€ steps up to 250€) around 20s, which is a huge improvement but still too slow if you want a boardroom like described in section 7.

6.3 Exact Bellman-Equation

A possible alternative to the former solution consists of an exact solving. After rearranging the equation as seen in 3, the problem can be solved directly and without iterations.

$$V(n,\vec{s}) = \max_{a\geq 0}\left\{\frac{\sum_i \pi(i,\vec{x}(a,\vec{s}))\cdot(\min(n,i)\cdot a - n\cdot l) + \sum_{i\neq 0}\pi(i,\vec{x}(a,\vec{s}))\cdot\delta\cdot V\left((n-i)^+,\vec{s}\right)}{1-\delta\cdot\pi(0,\vec{x}(a,\vec{s}))}\right\}$$

Figure 3: Exact Bellman-Equation

This approach enables a solving within a second which is fast enough for live price updates and a simulation of the selling process using the calculated prices.

6.4 Simulation sales based on the calculated prices

For vendors it is too risky to directly apply the automatic pricing strategy. Also evaluating whether the automatic strategy is superior to the manual, one would require a long testing phase and is again undesirable. To minimize the risk and get an impression before testing, vendors could check the prices for a market situation and compare with the manually selected price, but surely this requires too much work and would be too expensive.
To tackle this problem and to evaluate the automatic pricing strategy a simulation based on probabilities was built. For a given market situation and time interval, a random number equally distributed from 0 to 1 is chosen. Then the sales probabilities given the historical price and the calculated price are computed and compared to that random number. If the random number is smaller or equal to a sales probability, we have a sale for that price. As we only have very short time intervals and hence small probabilities, it is very unlikely to achieve a sale. To overcome this problem, we increased the timespans by factor 20, resulting in 2 days timespans, and increased the sales probabilities by factor 20. This is a valid approximation as it is much more likely to sell in a longer period than in a short one. The overall estimated profit can now be calculated by summing up the prices when a sale occurred and subtracting the holding costs over the complete time.

7. USER INTERFACE

As part of the implementation, a dashboard was created including the possibility of visualizing the historical price development for each book, different market situations whose input variables may be altered by the user as well as a profit estimation for a subset of books following the previous described pricing policy until they are sold out. Therefore, the user interface was structured into three major parts:

The first part enables the user to search through all available book items enriched by external information like the book cover, title, publisher and author. After choosing one book item, one is able to dive into available historical data which were put into graphs (see figure 4). Those historical

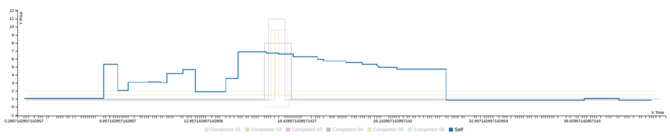

Figure 4: Historical pricing data of a book item

data include all competitor prices and settings as well as our own ones. Again, the data are enriched by external information like the publication and content details of the specific book item.

The second part of the user interface is the "boardroom", which facilitates a management perspective on the own market position. Here again, important details are accessible for each book item individual including competitor prices and one's own one. More importantly, the user is able to gain a generalized understanding of the market position. By resorting to the introduced bellman equation from the previous chapter (6), we simulate the new pricing strategy against the original one based on the historical data and visualize the overall revenue and more importantly the time until an sold out in order to reduce the warehouse charges and increase the profit (see figure 5).

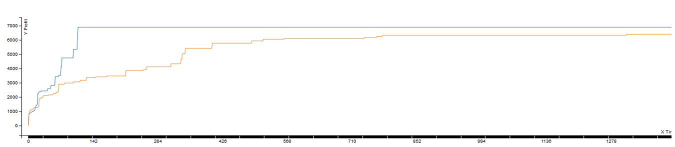

Figure 5: Boardroom

The last and third part is the "profit estimation" addressing the need of simulating and altering different market situation based on the historical data and estimating one's own profit for each of those market situation. Additionally, the user may not only alter the book item amount or his own price or rating but also the ones of all competitors for each single available book individual or generalized.

It is important to mention, that the user interface offers the opportunity to modify all settings including the competitors' prices at any stage of the data set to enable the user to alter the market situation retroactive for the purpose of simulation.

8. CONCLUSION

Within this elaboration we showed how to use logistic regression to estimate sales probabilities for books based on real data. Therefore, we discovered important features and explained ways to evaluate the quality of a trained model. Based on the sales probabilities, the bellman equation is used to predict the optimal prices in around a second. This is not only helpful for book vendors to instantly get an idea of how much they could charge for a book, but also the basis for automatic pricing strategies.

Historical sales data of a specific book can be visualized in our dashboard and compared against a simulation when using the estimated optimal prices. Even, if the model for the sales probabilities is not perfect yet, the pricing strategy using the calculated prices already shows superior quality in various situations.

We believe that automatic pricing strategies will be an essential part of the selling process in the future as it eliminates the need of setting prices manually and directly reacts on changing market situations. With the increasing number of competitors in the digital era and especially on platforms like Amazon or Ebay, automatic pricing strategies can be a competitive advantage and help increasing the revenue. We hope that we could demonstrate a successful application of one automatic pricing strategy for books and are convinced that these strategies are the way to go in nearly all domains.

The source code and the documentation will be available at
https://github.com/jaSunny/DynamicPricing
while screencasts are accessible under
https://www.youtube.com/watch?v=75dStkQiYNo,
https://www.youtube.com/watch?v=sdo328JU_0Y, and
https://www.youtube.com/watch?v=YJG9fGpJU_8.

9. ACKNOWLEDGMENTS

As part of this elaboration, special thanks goes to Dr. Rainer Schlosser and Martin Boissier for their continuous support and supervision. Also, we thank all attendant members of the EPIC research group for their fruitful discussions during the intermediate and final presentation.

References

[1] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.