Machine learning and intelligent optimization in tourism and hospitality: “Rocket Science” without the need of scientists

Roberto Battiti and Mauro Brunato, University of Trento, Italy

White paper, Feb 16, 2018

Abstract

Like most businesses, tourism and hospitality are undergoing a phase of disruptive innovation caused by the wider adoption of computers, networks and, above all, sophisticated and powerful **algorithms**. Algorithms automate business processes in a partial or total manner, by starting from repetitive and simple tasks but progressively reaching also more complex and “creative” tasks, traditionally associated with human decision making.

The concepts of “**automated creativity**” or “automated business innovation” sound like contradictions. We like to think that only human people can discover truly innovative ways of solving problems and radically improving business performance.

In this chapter we summarize two theoretical advancements in the past years which permit this disruptive innovation: **machine learning** and **intelligent optimization**.

Managers and decision makers reach decisions by some level of anticipation (expectation, prediction) of the effects of different choices. Even if not explicit, these decisions are based on a series of “What if?” questions, with answers given by expertise, gut feeling, or some level of logical and mathematical modelling.

Machine learning (**ML**), or **learning from data**, is a theory for deriving flexible models by starting only from the data produced by the business. The objective of these models is to *generalize* in a sound manner for cases not already encountered in the past.

After a model is available, computers can simulate the effects of millions of possible decisions, by predicting the output (for example the total profit of the hotel), and by creating and selecting one among the best decisions.

Intelligent optimization (**IO**) is this automated process of creating in an intelligent manner a large series of possible decisions, aiming at improving the current way of doing business.

In many practical cases, the objectives to reach are not crystal-clear from the beginning. A standard case is that of multiple-objectives optimization (e.g., maximizing customer satisfaction and minimizing costs). Fine-tuning the objectives by properly balancing different desiderata and constraints and by learning about the possibilities is the sign of intelligent managers. Machine learning represents an additional possibility, this time by learning and fine-tuning a proper global and combined objective to be achieved from the decision maker feedback.

Through machine learning and intelligent optimization, hotel managers have extremely powerful tools in their pockets to improve total profitability and customer satisfaction. It is up to them to understand the new possibilities (the overall vision), decide which possible changes they are considering (e.g., acting on prices, availability of different types, reservation rules, kind of offer, etc.), collect and organize the relevant data about the past performance, deliver them to **ML** tools to build models and run millions of software experiments via intelligent optimization (**IO**) to identify improving solutions.

Last but not least, the above procedure is not executed *una tantum*, but it can run in background by progressively collecting more and more fresh data about the business performance and by fine-tuning more and more the most appropriate decisions depending on the current market conditions, competition, changing customer preferences, etc.

The entire process executes in a pragmatic, never-sleeping, and never-ending helicoid of innovation and improvement. It is is based on scientific design of experiments and measurements, free from wishful thinking and ideologies. The current developments of machine learning and intelligent optimization make the whole process available also to medium-sized hotels, without requiring rocket scientists and engineers in the hotel team.

1 Machine Learning, and the surprising connections between Occam's razor and hotel management

Let's assume that an hotel manager needs to decide which discounts to apply to children of age less than 18. This problem requires filling a table with 18 values (one for each age, from 0 to 17) of the discount to apply (a percent value, from 0% —no discount at all— to 100% — children stay for free). When asking manager about why they selected *specific* discount values, answers range from "Why not?" to "We imitate what our neighbor is doing", to "We do not know, the software required some values and we had to fill them in in some way". Actually, determining an *optimal* solution for the above problem, one leading to the maximum possible profit, is not difficult because of laziness or ignorance but because of a simple counting argument.

There are 100 possible discount values for each age, to be multiplied to get an amazing 100^{18} total different ways of filling the complete table with percent values, that is, 1 followed by 36 zeros. The estimated age of the universe in seconds, assuming 13.7 billion years since the Big Bang, is 432,000,000,000,000,000. Even if one possible way of filling the discount table could be analyzed in each second, the entire life of the universe would permit to consider only a minuscule fraction of possibilities.

This explosion of possibilities is know also as the "**curse of dimensionality**". Smart optimization schemes can deliver optimal solutions in reasonable times in spite of enormous numbers of possibilities only for a selected list of special problems, which unfortunately excludes almost all complex problems of real-world interest. If hotel managers want to apply a scientific, principled manner to choose business settings, and not black magic, a much smaller number of possibilities has to be considered. For a simple example, the manager can decide among two possibilities, and set up an experiment to decide whether a 50% discount for children of all ages will improve profits w.r.t. no discount at all. Of course, a children discount may be desirable if the hotel targets families and wants to discourage couples or singles, but the decision should be based on measurable results. For example, an experiment can be done with *split testing* in the hotel website, or by trying the two possibilities for comparable analogous periods.

Similar conclusions hold for different decisions, like for setting proper prices for each type of room and for each day depending on demand, market evolution, competition, type of customer, groups, agencies, etc., problems which are much more difficult and more crucial than deciding about children discounts. Scientific approaches based on measurements (data) can be automated in part by using novel methods of machine learning and optimization, as we explain in the following discussion.

The above fact, related to the impossibility of dealing in an affective manner with decisions with too many parameters is deeply rooted in the scientific approach. A good scientific theory is characterized by brevity, and **concision is the source of power** enabling a solid scientific theory to conquer the world.

Occam's razor (a.k.a. "**law of parsimony**") is the problem-solving principle stating that in science the simplest theories have to be preferred. The idea is attributed to William of Ockham (c. 1287–1347), an English Franciscan friar and philosopher. The same "law of parsimony" is at the basis of machine learning (ML) or learning from data. Because the objective of ML is to generalize for new and unseen data, if the model is too complex, the examples presented during learning will be reproduced, but learning degrades to a sort of *superficial memorization* of the examples, not extracting the deep underlying rules.

For revenue management, a crucial aspect is to forecast demand by customers and hotel occupation levels. Data about previous months and years, together with data from similar hotels, can be precious to "machine learn" a forecast. Again simple models with a small number of free parameters need to be used to predict the future (before it happens).

For a second concrete example, let's consider a simplified case in hotel revenue management (G. Bitran & Caldentey, 2003). A manager has to decide the proper price for selling a room of his new hotel through an online booking engine. Customers visit the hotel website, get offers and decide whether to book or not. For this example, let's imagine that the interaction is one-shot.

The manager (let's call him a decision maker or DM) wants to optimize the profit obtained from the sale opportunity. Two sources of knowledge influence the choice about the proper price: a model of hotel costs and a model about acceptance of the offer by the customer.

Let's start from the first source, and look at Fig. 1. The **model of hotel costs** can be obtained via a traditional "design from specifications" approach. A proper approximation for this short-term decision is to consider only the *marginal cost*, that is, the cost caused by selling one more room (cleaning, electricity and hot water consumed by the customer, if any). This model of hotel costs can be visualized by a graph of profit as a function of the price of sale. For sure, the price has better

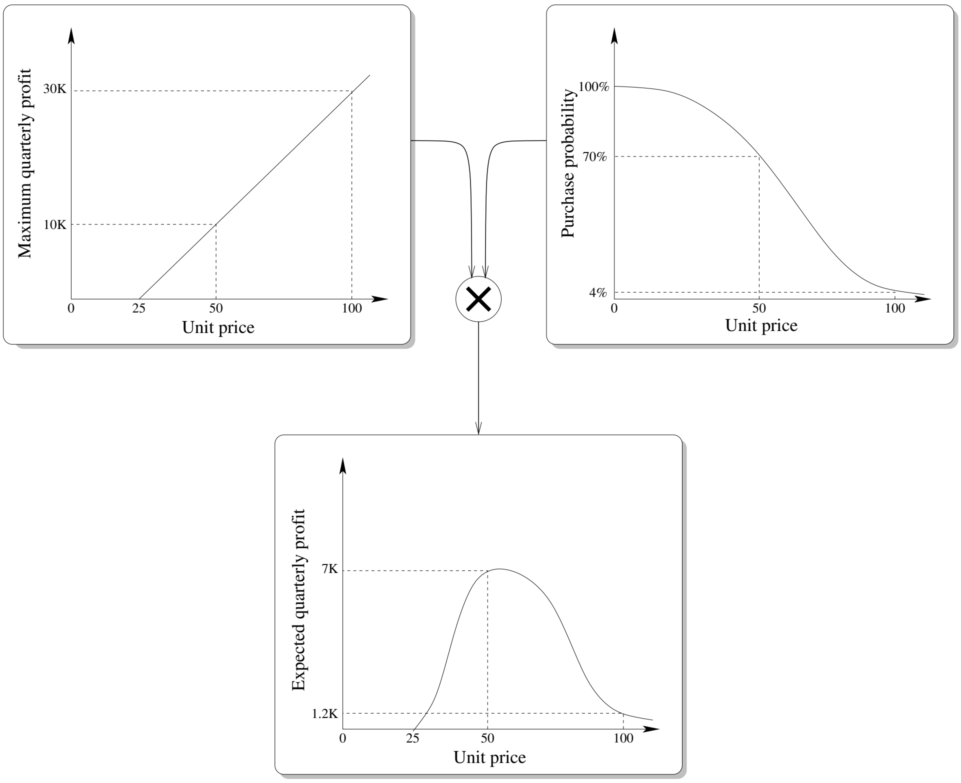

Fig. 1: Top left: above the break-even price (25 in this example), the marginal profit is proportional to sale price increase; however, by taking into account a user acceptance model (top right), we can more realistically forecast our expected profit by multiplying the two (bottom) and locating the optimal sale price (slightly above 50 in this example).

be higher than the marginal cost (otherwise the hotel will lose money). After this point, the higher the price, the higher the profit. But a wise manager knows that if the price is exaggerated, the customer will not accept the offer.

This fact leads to second source of knowledge: a **model of acceptance of the offer by the customer**. Here the "design from specifications" approach shows all its shortcomings. Let's assume one wants to estimate the probability of a customer accepting the offer as a function of the price. One could proceed through interviews, through studies of the competition, through consultants, but for sure one will never reach the clean "designed" formulation comparable to the formula "additional profit equals price minus marginal costs" of the previous model.

Machine learning (a.k.a. learning from data) comes to the rescue when one does not have an analytic solution, but one has data to construct an empirical solution (Battiti & Brunato, 2018),(Abu-Mostafa, Magdon-Ismail, & Lin, 2012). In the example, one can design an experiment, randomly produce prices in a give (reasonable) range and estimate from the actual customer events the probability of acceptance of an offer as a function of price.

After automatically building this second model one is ready to use (automated) **optimization**. The expected profit obtained as a function of a specific price will be given by the multiplication of the cost model by the probability of acceptance. To identify the optimal price, a **software simulator** can try the different prices, calculate the expected profit and output the best possible price, the one which maximizes profit.

In a full fledged real-world situation the parameters to be chosen by the DM will be more than one, so that the DM should use intelligent optimization techniques, much smarter than the above mentioned exhaustive consideration of all possible choices. In addition, the objective to improve can be much more complex than profit maximization. Other objectives like customer satisfaction, long-term profitability, compliance with regulations, etc. have to be taken into account. The less clear the initial objectives are, the more iterative the process becomes. The DM will experiment with initial goals (objectives) but then continuously monitor results, fine-tune and refine goals, define different tradeoffs etc., in a never-ending process of continuous innovation and improvement. Even in this case of complex and changing goals, ML can learn from judgments by the DM and define better and clearer objectives for optimization.

How is it possible to learn (only) from the data and accomplish challenging tasks in an automated, algorithmic manner? If you are new to this area, a proof of existence of this kind of data-based learning solutions is our brain. Learning to ride a bicycle is an incredibly challenging task if approached with the traditional technique of "explicit design from specifications". It looks like one should first master the laws of physics, classical mechanics, kinematics, dynamic analysis, forces, accelerations, law of gravitation, etc. After mastering the basic laws, one should then design rules like "if wheels have these angles, velocity is this, etc. apply a force of six Newtons to the right pedal, etc." Surprisingly enough, kids can learn to drive a bicycle with nothing but a very fuzzy guidance by parents (sit, put your feet on the pedals, look ahead,...) some initial trials, and possibly a couple of falls and some bruises. After this quick "machine learning" phase they can glide along without touching their feet down to correct themselves, and they are ready to begin pedaling. No theory, no logical thinking, no design, just data and feedback collected by our brain and progressively changing strengths of connections between neurons (called synapses). If your substitute the bike with your hotel and the data collected by your neurons with data collected by your business (prices, offers, reviews, costs, ...) you can use similar "sub-symbolic learning algorithms" to drive your business to an ever increasing success and profitability.

In the following we briefly mention some theoretical tools that can be used for tourism and hospitality, with a presentation of the basic principles, of some concrete cases, and with references to the more detailed and growing literature on this topic. The book (Battiti & Brunato, 2018) can be consulted for a much deeper and extended introduction to the different "learning and intelligent optimization" methods.

2 Supervised machine learning

Let's imagine that the objective is to model the probability that a customer accepts an offer, a crucial ingredient in setting proper prices and proper offer characteristics.

A model of this kind can be represented by a "computational box". It receives as input the characteristics of the offer (room characteristics like number of beds, square meters, etc., offer rules like deposit, cancellation rules, etc.) and calculates as output the probability of acceptance. The output will depend on the inputs but also on the values of the internal parameters of the computational box.

In supervised machine leaning one starts from data consisting of a set of input-output pairs (collected from the results of previous offers accepted or rejected by customers) and aims at setting (learning) the internal parameters of the flexible model to reproduce the desired outputs and generalize to new inputs.

To fix the notation, a training set of ℓ *tuples* (ordered lists of elements) is considered, where each tuple is of the form $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, \ell$; $\mathbf{x}_i$ being a vector (array) of input parameter values in d dimensions ($\mathbf{x}_i \in R^d$); y_i being the measured outcome to be learned by the algorithm. In **regression**, the output is a real number, and the objective is to model the relationship between a dependent variable (the output y) and one or more independent variables (the input features $\mathbf{x}$).

Supervised learning uses the examples to build an association (a function) $y = \hat{f}(\mathbf{x})$ between input $\mathbf{x}$ and output y. The association is selected within a **flexible model** $\hat{f}(\mathbf{x}; \mathbf{w})$, where the flexibility is given by some **tunable parameters** (or **weights**) $\mathbf{w}$.

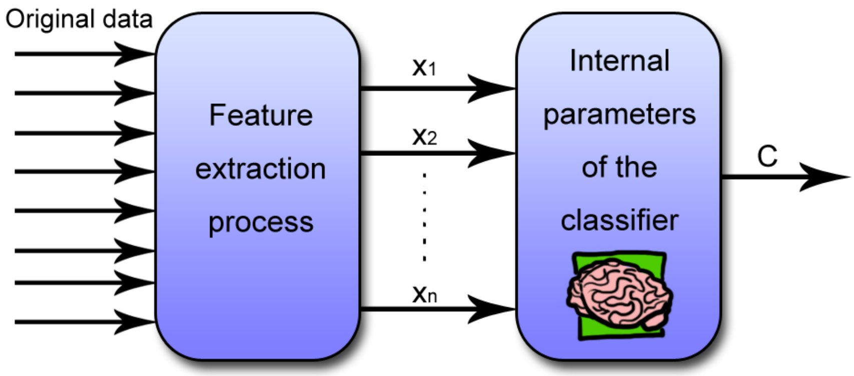

Fig. 2: Supervised learning architecture: feature extraction and classification.

A scheme of the architecture is shown in Fig. 2, the two parts of feature extraction and identification of optimal internal weights of the classifier are distinguished. In many cases feature extraction requires some human insight, while the **determi-**

nation of the best parameters is fully automated, this is why the method is called *machine* learning after all. The free parameters are fixed by **demanding that the learned model works correctly on the examples** in the *training set*.

A suitable *error measure* is the sum of the errors between the correct answer (given by the example label) and the outcome predicted by the model (the output ejected by the multi-purpose box). If the error is zero, the model works correctly on the given examples. The smaller the error, the better the average behavior on the examples.

Supervised learning therefore becomes **minimization of a specific error function**, depending on parameters $\mathbf{w}$. The brute-force method to find proper values of the internal parameters $\mathbf{w}$ can be to try all possible values (after deciding a suitable number of levels), measure the error function and pick the parameters leading to the minimum error. Brute-force rapidly leads to excessive computing times but mathematical optimization presents a series of smarter ways to determine optimal values (or approximations thereof).

Minimization of an error function is a first critical component to achieve a higher level of automation, but not the only one. If the **model complexity** (the flexibility, the number of tunable parameters) is too large, learning the examples with zero errors becomes trivial, but predicting outputs for new data may fail brutally. In the human metaphor, if learning becomes rigid memorization of the examples without grasping the underlying model, students have difficulties in generalizing to new cases.

When learning from labeled examples one needs to follow **careful experimental procedures** to measure the effectiveness of the learning process. In particular, it is a mistake to evaluate the performance of the learning systems on the same examples used for training. The objective of machine learning is to obtain a system capable of **generalizing** to new and previously unseen data. Otherwise the system is not learning, it is merely memorizing a set of known patterns.

Let's assume we have a *supervisor* (a software program or an experimental process) who can generate labeled examples with a given probability distribution. In tourism a concrete supervisor can be the software behind the website of the hotel, collecting user requests, offers and the corresponding output (acceptance or rejection by the customer).

In general, one should ask the supervisor for some examples during training, and then test the performance by asking for some fresh examples. Ideally, the number of examples used for training should be sufficiently large to permit convergence, and the number used for testing should be very large to ensure a statistically sound estimation.

This ideal situation may be far from reality. In some cases the set of examples is rather small, and has to be used in the best possible way *both* for training *and* for measuring performance. In this case the set has to be clearly partitioned between a **training set** and a **validation set**, the first used to train, the second to measure performance, as illustrated in Fig. 3.

In general, the learning process fixes the model parameters to make the model

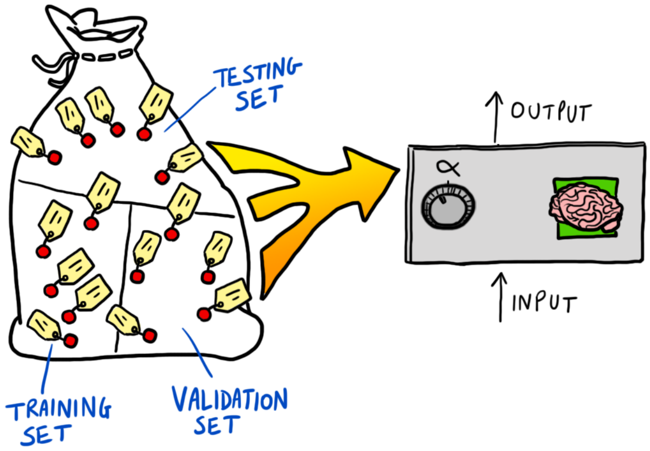

Fig. 3: Labeled examples need to be split into training, validation and test sets.

reproduce the output of the training data as well as possible. If we then take an independent sample of validation data from the same population as the training data, it will generally turn out that the error on the validation set will be larger than the error on the training set. This discrepancy is likely to become severe if training is excessive, leading to **over-fitting** (**overtraining**), likely to happen when the number of training examples is small, or when the number of parameters in the model is large.

Let's mention some notable examples of applications of machine learning techniques, including methods based on statistics, to tourism and hospitality.

Revenue Management is based upon demand modeling and forecasting, which provide the essential input for the optimization model. Early RM setups assumed the "independent demand model", i.e., assumed that demand is independent for different products or price classes (Talluri & Van Ryzin, 2006). This is a rough approximation, which excludes any substitution effects by consumers. If a business person has to book a room, and expensive business rooms are offered together with cheaper "leisure" rooms, it is obvious that some of them will decide to opt for the cheaper rooms (i.e., demand for expensive room types will depend also on availability of cheaper room types). Modelling how customers decide among a given offer set (a set of rooms types offered by the hotel) is a crucial ingredient in order to obtain more realistic results, and better final profits. A *choice model* can be thought of as a conditional probability distribution that for any offer set yields the probability that an arriving customer purchases a given product in that set. An early work to consider choice behavior in networks is (P. Belobaba & Hopperstad, 1999). Recently, (Ratliff, Rao, Narayan, & Yellepeddi, 2008),(Vulcano, Van Ryzin, & Chaar, 2010) demonstrated improvements from using multinomial logit (MNL) choice-based RM approaches in the airline market. Maximum Likelihood Estimation with incomplete data (only purchase transaction data, while no-purchase outcomes are unobservable) is used to derive model parameters from the data. Stochastic steepest ascent is used in simulation-based optimization.

Most of the past research work has focused on devising optimal decisions given a choice model. In most cases the choice model is assumed to be given and of the MNL type. The MNL model is popular because it is *tractable* both in terms of estimating its parameters and solving decision problems, but is has known deficiencies like the Independence of Irrelevant Alternatives (IIA) property (Ben-Akiva, Lerman, & Lerman, 1985).

A non-parametric approach to modeling choice is proposed in (Farias, Jagabathula, & Shah, 2013). The objective is to use of historical sales data in the prediction of revenues or sales from offering a particular assortment of products (predicting the conversion-rate for an offer set).

The work in (Gallego, Ratliff, & Shebalov, 2014) considers approximations to render the choice-based network RM problems tractable to solve (the initial formulation has an exponential number of variables). A recent review of Choice-based Revenue Management is (Strauss, Klein, & Steinhardt, 2018). An Artificial Neural Network-based technique for on-line hotel booking is proposed in (Corazza, Fasano, & Mason, 2014), which mentions the problem of obtaining appropriate training examples: past outputs given by human operators is not necessarily optimal

A comparison of forecasting schemes is (Weatherford & Kimes, 2003). Time series forecasting with trigonometric models is considered in (Pereira, 2016)

Up to now we considered supervised ML, but different methods of machine learning are available for different contexts. A notable case is **unsupervised machine learning**. In this context only example of input data are available, with no associated output label. The objective is to explore the data to discover interesting structure. As an example, one is given data about previous customers of an hotel (sex, age, country of origin, money spent at the hotel, history of the interaction with the hotel or with the hotel website, data from social profiles like Facebook, etc.) and the objective is to identify meaningful clusters to be used for future marketing efforts. A detailed review of unsupervised machine learning methods is present in (Battiti & Brunato, 2018).

Possible applications of clustering techniques are for the dynamic identification of online virtual communities (VC), groups of people trying to achieve certain purposes, with a similar interest. They are defined as “aggregations of individuals or business partners who interact around a shared interest, where the interaction is at least partially supported and/or mediated by technology and guided by some protocols or norms” (Porter, 2004). Marketing efforts are often based on customer segmentation (or clustering), which can be obtained by unsupervised ML.

3 Intelligent optimization

An hotel manager faces two kinds of crucial activities. A first activity is to choose among a finite set of given possibilities, imagine picking one among a set of online sales channels (booking.com, Tripadvisor, Expedia?). The alternatives are given, the task is to compare them against business objectives to pick the best one in a rational manner.

A second and more challenging activity requires **designing new and improving solutions among a (practically) infinite set of possibilities**. The possibilities are not given in a finite list, the task requires the design and construction of possible solutions, with creativity and innovation.

If the "sensible choice" among a finite and limited list of possibilities is a minimal competence required by a professional manager, the second activity characterizes the most successful and visionary decision makers. Unfortunately, the second creative activity is not trivial. The fact that alternatives are infinite requires a smart creation and test of a few successful combinations.

Smart algorithms, when properly designed, directed and inserted into the business practice are becoming the necessary step for a consistent improvement. The success of an innovation effort will be based more on scientific principles than on the luck of encountering a few successful solutions.

After **a computable model of the activity** is available, for example via machine learning from data, the effect of a particular selection of relevant business variables (e.g. prices in time, offers, types of rooms, reservation and cancellation rules, discounts for selected customer groups) can be estimated. If a human decision maker can analyze a couple of possible choices per day, a computer, or even a series of computers running in parallel, can analyze millions or billions of possibilities, by deriving averages, error bars (or distribution of possible outcomes) and objective output measures to be used in determining an improving configuration (e.g. a pricing policy leading to a larger profit and/or better customer satisfaction). In **simulation-based optimization**, already used in many engineering areas, a simulator of the system (e.g. a simulator to derive the annual profit of an hotel) is run starting from different settings of the possible choices generated in an intelligent manner by the optimization algorithm.

There is little doubt that algorithms and computational speed will beat even the more expert human person in the long term. Nonetheless, computers also have limits. In fact the theory of *bounded rationality* by Herbert A. Simon (Nobel Prize in Economics 1978) was actually based more on his knowledge of computer science than on his knowledge of human nature. The term "bounded rationality" designates rational choice that takes into account the cognitive limitations of both knowledge and cognitive capacity. The history of the last decades of theoretical computer science abound with negative results about the possibilities to solve in an optimal manner, or even to approximate, many problems (notably to NP-hard problems) in acceptable (polynomial) computing times, in the worst case. When the number of possibilities to be examined grows in an exponential manner, computers have to abandon exhaustive techniques that consider all possibilities in favor of smarter heuristic techniques that consider only a small but interesting subset of possibilities. The assurance about getting the best possible solution must be abandoned and one has to be satisfied with a solution that improves the current state.

As an example of this complexity barrier encountered in tourism and hospitality, consider the task of setting prices for different kinds of reservations. If one has to fill a huge table with a price value for every day in the calendar, for every number of days in advance (w.r.t. checking time), for 10 room types, for an occupation ranging from 1 to 4 beds, for kids of different ages, for different cancellation rules, etc. one has already an unmanageable number of possibilities to consider.

Many issues in everyday's life are related to solving optimization problems, to improve solutions, or to find solutions which are so good that no other solution is better (called "global optima"). In very rare special cases algorithms can deliver an optimal solution in reasonable computing times, in other cases, including almost all real-world problems, one has to resort to heuristics. In many practical cases these heuristics, although without guarantees of optimality, can deliver solutions which are in practice very effective, improving on the state of the art, in some cases hardly distinguishable from the theoretically optimal ones (estimation and measurements errors can easily cancel small theoretical differences).

In abstract and general terms, in optimization one is given a function f defined on a set of possible input values $\mathcal{X}$. The function $f(\mathcal{X})$ to be optimized is called with more poetic names in some communities: *fitness* function, *goodness* function, *objective* function. If $\mathcal{X}$ is defined by a discrete set of possibilities (like binary values, permutations, integers) one speaks about **discrete optimization**. On the contrary, **continuous optimization** considers real-valued inputs.

One aims at finding the input configuration leading to the least possible value of the function f. Often a set of **constraints** on $\mathcal{X}$ have to be satisfied for a solution to be considered **admissible**.

In an hotel, f can be the annual profit from the hotel, and $\mathcal{X}$ can be the set of all possible configurations of variables influencing the profit (prices, offers, booking rules, overbooking strategies, ...) which can and will be decided by the manager.

Mathematical optimization techniques presents a series of possible methods to build heuristics. We are not presenting here special optimization problems which can be solved exactly like Linear and Quadratic Programming, although LP and QP are extremely useful and widely used, they assume that the problem has a very special linear or quadratic form. If you are lucky to encounter one of this problems, there is an abundance of software which can be used. We concentrate on the more frequent case in which the context and the problems to be solved do not have this very particular structure. For sure this is a frequent case when the flexible model is derived from noisy and non-linear business data through machine learning, or from the results of running a complex hotel simulator.

Optimization techniques are heavily used in Revenue Management, which flourished after the deregulation of the airline industry in the 1970s. An approximated network flow formulation for network revenue management is proposed in

(Glover, Glover, Lorenzo, & McMillan, 1982). (P. P. Belobaba, 1989) developed the EMSR (Expected Marginal Seat Revenue) heuristic to determine booking limits for the different fare classes (using Littlewood's equation (Littlewood, 1972)). (Williamson, 1992) used simulation to address the network aspect of RM. The idea of SBO and Monte Carlo integration in the RM context is advocated in (Robinson, 1995) for the single-leg problem.

A seminal work on optimal dynamic pricing of inventories with stochastic demand over a finite horizons is (Gallego & Van Ryzin, 1994). RM in the context of e-commerce for dynamic automated sales is studied in (Boyd & Bilegan, 2003). Revenue management and e-commerce in the airline industry is analyzed in (Boyd & Bilegan, 2003)

Simulation-Based booking limits for RM in the airline business are studied in (Bertsimas & De Boer, 2005), and later extended by (Van Ryzin & Vulcano, 2008), which considers a continuous model, less realistic in terms of the fine-grained details but leading to an easier continuous optimization problem.

Simulation-based optimization and a stochastic steepest ascent algorithm is used in (van Ryzin & Vulcano, 2008) to maximize revenue under customer choice behavior. Their simulation-based method optimizes over the parameters of a virtual nesting control policy. According to the authors modeling **flexibility** is a plus: "one can make essentially arbitrary changes in the model of demand and customer behavior without impacting the way the optimization algorithm functions." (Chaneton & Vulcano, 2011) present a stochastic gradient algorithm for improvement of bid prices with customer choice. (Klein, 2007) introduces auto-adaptive bid prices by means of the metaheuristic **scatter search**, assuming independent demand.

The recent work (Ayvaz-Cavdaroglu, Gauri, & Webster, 2017) addresses dynamic pricing in the Cruise Line Industry, allowing for partial substitutability among products and including also marketing expenses. The optimal solution of the quasiconvex optimization problem is obtained by Karush-Kuhn-Tucker(KKT) conditions.

Dynamic pricing with strategic customers is considered in (Gönsch, Klein, Neugebauer, & Steinhardt, 2013), simultaneously learning and optimizing in (den Boer & Zwart, 2013), a survey of dynamic pricing with learning in (den Boer, 2015)

Risk is considered in RM in (Feng & Xiao, 2008). *Robust RM* assumes that exact distributional information is not available and uses uncertainty sets to describe ranges, for example for demand realizations or distribution parameters. Numerical methods to construct the uncertainty set from historical data are used in (Sierag & van der Mei, 2016). Risk-sensitive capacity control in revenue management is studied in (Barz & Waldmann, 2007). Risk in revenue management and dynamic pricing is considered in (Levin, McGill, & Nediak, 2008), which also provides approximation techniques as the optimal dynamic pricing policy may be hard to compute.

Least squares approximate policy iteration for learning bid prices in choice-based RM is proposed in (Koch, 2017) A survey on risk-averse and robust revenue management is present in (Gönsch, 2017) Reinforcement learning for RM is considered in (Gosavi, 2004)

(Ling, Dong, Guo, & Liang, 2015) optimizes hotel room availability by allocating rooms between the distribution channels of the hotel and OTA. An empirical study of dynamic pricing in hotels is (Abrate & Viglia, 2016), which evaluates the importance of each factor to determine the measured prices. Recent developments in dynamic pricing are presented in (Chen & Chen, 2015).

4 Simulation-based optimization for hotels

As mentioned, a scientific approach to hotel management requires measurements. In particular, measurements are needed to evaluate the effect of business choices (e.g., prices and reservation rules) on the total business profit.

Unfortunately, real experiments with hotels require very long times (of the order of months or years) and risk disrupting the daily activities of the hotel (hotel guests may object to being the subject of experiments). But software and powerful computers permit a less disruptive and much faster possibility: to design a simulator of a specific hotel, and then to **use the simulator instead of the real hotel**. With a simulator, a year in the hotel life can be simulated in seconds, which means that thousands (or even millions) of possible choices can be analyzed in reasonable times, a number which greatly extends the human possibilities of analysis, which are usually limited to a handful of possibilities.

Needless to say, the simulator needs to be “trained” to reproduce the real hotel with a high level of fidelity, in order to be trusted, but this is the typical context of machine learning for which a growing number of theoretical tools and software is now available.

Among the fist systematic usages of experimentation with a simulation model of the reservation process is (Smith, Leimkuhler, & Darrow, 1992). The use of simulation in hospitality for teaching, practice, and research purposes is advocated in (Thompson & Verma, 2003). Simulation is the main tool currently used for hotel revenue management problems of multi-night stays. (Weatherford, 1995) and (G. R. Bitran & Mondschein, 1995) use simulation models and heuristics with data from hotels to define accept/reject decisions for booking, also for the case of different length-of-stay and multiple types of rooms with downgrading (they note that “the computational burden becomes enormous, when real size problems with several types of rooms are considered”). Their work has been extended in (Baker, 1994). Stochastic approaches and robust optimization for RM in hotels are considered in (Lai & Ng, 2005) and (Liu, Lai, Dong, & Wang, 2006).

Simulation-based optimization is used in (Koch, Gönsch, Hassler, & Klein, 2016) to derive practical decision rules for risk-averse revenue management. Simulation optimization for revenue management of airlines with cancellations and overbooking is discussed in (Gosavi, Ozkaya, & Kahraman, 2007), which also uses Simulated Annealing. According to them, simulation can **easily accommodate realistic assumptions** which render theoretical models intractable and **do not require detailed knowledge** of the internal structure of the stochastic system.

(Subulan, Baykasoğlu, Akyol, & Yildiz, 2017) uses simulation optimization and meta-heuristics for network revenue management in airlines (accept/reject decisions). A general overview of SBO can be found in (Gosavi, 2015; Spall, 2003).

4.1 Reactive Search Optimization (RSO): Learning While Optimizing

Even if the initial optimization problem is black-box (one knows only inputs and associated output, not the internal structure of the box), the more points are generated in input space and evaluated, the more knowledge is accumulated, in implicit form. Data about the past history of the search can be exploited to generate internal explicit models and improve the efficiency and effectiveness of the future optimization effort. In a way, RSO tends towards truly intelligent problem-solving machines, which learn and self-improve the more they work, in a way similar to humans, or similar to reactive biological systems. Think about the lifelong learning of a violinist, from the first mechanical and "symbolic" rule-based movements, to the real mastery of a Paganini.

Reactive Search Optimization (RSO) advocates the integration of online machine learning techniques into optimization heuristics. The word reactive hints at a ready response to events during the search through an internal feedback loop for online self-tuning and dynamic adaptation. In RSO the past history of the search and the knowledge accumulated while moving in the configuration space is used for self-adaptation in an automated manner: the algorithm maintains the internal flexibility needed to address different situations during the search, but the adaptation is automated, and executed while the algorithm runs on a single instance and reflects on its past experience. Machine learning is therefore an essential ingredient in the RSO soup, as illustrated in Fig. 4.

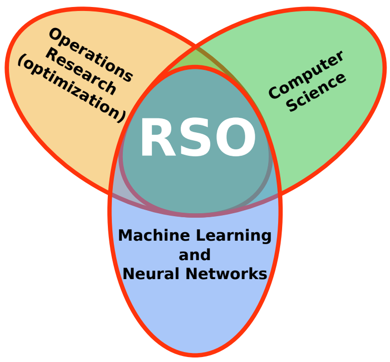

Fig. 4: RSO is at the intersection of optimization, computer science (algorithms and data structures) and machine learning.

Many problem-solving methods are characterized by a certain number of choices and free parameters, whose appropriate setting and tuning is complex. In some cases the parameters are tuned through a feedback loop that includes the user as a crucial learning component: different options are developed and tested until acceptable results are obtained. The quality of results is not automatically transferred to different instances and the feedback loop can require a slow "trial and error" process when the algorithm has to be tuned for a specific application. In Machine Learning a rich variety of "design principles" is available that can be used in the area of parameter tuning and optimal choice for heuristics. The lack of human intervention does not imply higher unemployment rates for researchers. On the contrary, one is now loaded with a heavier task: the algorithm developer must transfer his intelligent expertise into the algorithm itself, a task that requires the exhaustive description of the tuning phase in the algorithm.

Seminal papers related to using memory in a strategic manner to guide heuristics to continue exploration beyond local minima are the ones by Fred Glover about tabu search, scatter search and path relinking, and related metaheuristics, see for example (Glover, 1990), (Glover, Laguna, & Martí, 2000) and the amusing (Glover, 2007). Other inspiring papers that you may want to read to get a taste of related topics are (Moscato, 1989) about memetic algorithms, (Yao, 2002) about evolving neural networks, and (Blum & Roli, 2003) about meta-heuristics.

5 Conclusion

Machine learning and intelligent optimization are driving the latest "automation revolution" in most businesses. The tourism and hospitality sector is not an exception and is being radically transformed by this wave of intelligent algorithms.

Although based on recent theoretical advances of a high level of mathematical complexity, a positive fact is that the daily application of these tools does not require hotels to have scientists and engineers on their team. On the contrary, after a suitable design of the processes and after connecting relevant business data, the entire modeling and optimization process can run in the background, periodically delivering suggestions for improving profitability or periodically modifying some business parameters (like prices).

There is little doubt that the combination of human intelligence with the speed, modeling and simulation capabilities of computers will provide a competitive edge with respect to more traditional ways of managing hotels.

References

Abrate, G., & Viglia, G. (2016). Strategic and tactical price decisions in hotel revenue management. *Tourism Management*, *55*, 123–132.

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012). *Learning from data* (Vol. 4). AMLBook New York, NY, USA:.

Ayvaz-Cavdaroglu, N., Gauri, D. K., & Webster, S. (2017). Empirical evidence of revenue management in the cruise line industry. *Journal of Travel Research*, 1–17.

Baker, T. K. (1994). *New approaches to yield management: comprehensive overbooking/allocation heuristics for the hotel industry* (Unpublished doctoral dissertation). The Ohio State University.

Barz, C., & Waldmann, K.-H. (2007). Risk-sensitive capacity control in revenue management. *Mathematical Methods of Operations Research*, *65*(3), 565–579.

Battiti, R., & Brunato, M. (2018). *The lion way. machine learning* plus *intelligent optimization*. LIONlab, University of Trento, Italy. Retrieved from `http://intelligent-optimization.org/LIONbook/`

Belobaba, P., & Hopperstad, C. (1999, April). Mit simulation study: Pods results update. boeing. In *1999 agifors reservations and yield management study group symposium.*

Belobaba, P. P. (1989). Or practice—application of a probabilistic decision model to airline seat inventory control. *Operations Research*, *37*(2), 183–197.

Ben-Akiva, M. E., Lerman, S. R., & Lerman, S. R. (1985). *Discrete choice analysis: theory and application to travel demand* (Vol. 9). MIT press.

Bertsimas, D., & De Boer, S. (2005). Simulation-based booking limits for airline revenue management. *Operations Research*, *53*(1), 90–106.

Bitran, G., & Caldentey, R. (2003). An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, *5*(3), 203–229.

Bitran, G. R., & Mondschein, S. V. (1995). An application of yield management to the hotel industry considering multiple day stays. *Operations research*, *43*(3), 427–443.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, *35*(3), 268–308.

Boyd, E. A., & Bilegan, I. C. (2003). Revenue management and e-commerce. *Management Science*, *49*(10), 1363–1386.

Chaneton, J. M., & Vulcano, G. (2011). Computing bid prices for revenue management under customer choice behavior. *Manufacturing & Service Operations Management*, *13*(4), 452–470.

Chen, M., & Chen, Z.-L. (2015). Recent developments in dynamic pricing research: multiple products, competition, and limited demand information. *Production and Operations Management*, *24*(5), 704–731.

Corazza, M., Fasano, G., & Mason, F. (2014). An artificial neural network-based technique for on-line hotel booking. *Procedia Economics and Finance*, *15*, 45–55.

den Boer, A. V. (2015). Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, *20*(1), 1–18.

den Boer, A. V., & Zwart, B. (2013). Simultaneously learning and optimizing using controlled variance pricing. *Management science*, *60*(3), 770–783.

Farias, V. F., Jagabathula, S., & Shah, D. (2013). A nonparametric approach to modeling choice with limited data. *Management Science*, *59*(2), 305–322.

Feng, Y., & Xiao, B. (2008). A risk-sensitive model for managing perishable products. *Operations Research*, *56*(5), 1305–1311.

Gallego, G., Ratliff, R., & Shebalov, S. (2014). A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research*, *63*(1), 212–232.

Gallego, G., & Van Ryzin, G. (1994). Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management science*, *40*(8), 999–1020.

Glover, F. (1990). Tabu search, Part I1. *ORSA journal on Computing*, *2*(1), 4–32.

Glover, F. (2007). Tabu search — Uncharted domains. *Annals of Operations Research*, *149*(1), 89–98.

Glover, F., Glover, R., Lorenzo, J., & McMillan, C. (1982). The passenger-mix problem in the scheduled airlines. *Interfaces*, *12*(3), 73–80.

Glover, F., Laguna, M., & Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, *39*(3), 653–684.

Gönsch, J. (2017). A survey on risk-averse and robust revenue management. *European Journal of Operational Research*, *263*(2), 337–348.

Gönsch, J., Klein, R., Neugebauer, M., & Steinhardt, C. (2013). Dynamic pricing with strategic customers. *Journal of Business Economics*, *83*(5), 505–549.

Gosavi, A. (2004). A reinforcement learning algorithm based on policy iteration for average reward: Empirical results with yield management and convergence analysis. *Machine Learning*, *55*(1), 5–29.

Gosavi, A. (2015). Simulation-based optimization: an overview. In *Simulation-based optimization* (pp. 29–35). Springer.

Gosavi, A., Ozkaya, E., & Kahraman, A. F. (2007). Simulation optimization for revenue management of airlines with cancellations and overbooking. *Or Spectrum*, *29*(1), 21–38.

Klein, R. (2007). Network capacity control using self-adjusting bid-prices. *OR Spectrum*, *29*(1), 39–60.

Koch, S. (2017). Least squares approximate policy iteration for learning bid prices in choice-based revenue management. *Computers & Operations Research*, *77*, 240–253.

Koch, S., Gönsch, J., Hassler, M., & Klein, R. (2016). Practical decision rules for risk-averse revenue management using simulation-based optimization. *Journal of Revenue and Pricing Management*, *15*(6), 468–487.

Lai, K.-K., & Ng, W.-L. (2005). A stochastic approach to hotel revenue optimization. *Computers & Operations Research*, *32*(5), 1059–1072.

Levin, Y., McGill, J., & Nediak, M. (2008). Risk in revenue management and dynamic pricing. *Operations Research*, *56*(2), 326–343.

Ling, L., Dong, Y., Guo, X., & Liang, L. (2015). Availability management of hotel rooms under cooperation with online travel agencies. *International Journal of Hospitality Management*, *50*, 145–152.

Littlewood, K. (1972). Forecasting and control of passenger bookings. *Airline Group International Federation of Operational Research Societies Proceedings, 1972*, *12*, 95–117.

Liu, S., Lai, K. K., Dong, J., & Wang, S.-Y. (2006). A stochastic approach to hotel revenue management considering multiple-day stays. *International Journal of Information Technology & Decision Making*, *5*(03), 545–556.

Moscato, P. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, *826*.

Pereira, L. N. (2016). An introduction to helpful forecasting methods for hotel revenue management. *International Journal of Hospitality Management*, *58*, 13–23.

Porter, C. E. (2004). A typology of virtual communities: A multi-disciplinary foundation for future research. *Journal of Computer-Mediated Communication*, *10*(1), 00–00.

Ratliff, R. M., Rao, B. V., Narayan, C. P., & Yellepeddi, K. (2008). A multi-flight recapture heuristic for estimating unconstrained demand from airline bookings. *Journal of Revenue and Pricing Management*, *7*(2), 153–171.

Robinson, L. W. (1995). Optimal and approximate control policies for airline booking with sequential nonmonotonic fare classes. *Operations Research*, *43*(2), 252–263.

Sierag, D., & van der Mei, R. (2016). Single-leg choice-based revenue management: a robust optimisation approach. *Journal of Revenue and Pricing Management*, *15*(6), 454–467.

Smith, B. C., Leimkuhler, J. F., & Darrow, R. M. (1992). Yield management at american airlines. *interfaces*, *22*(1), 8–31.

Spall, J. (2003). Introduction to stochastic searchand optimization: Estimation, simulation and control. *Wiley*, *34*, 54–58.

Strauss, A. K., Klein, R., & Steinhardt, C. (2018). A review of choice-based revenue management: Theory and methods. *European Journal of Operational Research*.

Subulan, K., Baykasoğlu, A., Akyol, D. E., & Yildiz, G. (2017). Metaheuristic-based simulation optimization approach to network revenue management with an improved self-adjusting bid price function. *The Engineering Economist*, *62*(1), 3–32.

Talluri, K. T., & Van Ryzin, G. J. (2006). *The theory and practice of revenue management* (Vol. 68). Springer Science & Business Media.

Thompson, G. M., & Verma, R. (2003). Computer simulation in hospitality teaching, practice, and research. *Cornell Hotel and Restaurant Administration Quarterly*, *44*(2), 85–93.

van Ryzin, G., & Vulcano, G. (2008). Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing & Service Operations Management*, *10*(3), 448–467.

Van Ryzin, G., & Vulcano, G. (2008). Simulation-based optimization of virtual nesting controls for network revenue management. *Operations Research*, *56*(4), 865–880.

Vulcano, G., Van Ryzin, G., & Chaar, W. (2010). Om practice—choice-based revenue management: An empirical study of estimation and optimization. *Manufacturing & Service Operations Management*, *12*(3), 371–392.

Weatherford, L. R. (1995). Length of stay heuristics: do they really make a difference? *Cornell Hotel and Restaurant Administration Quarterly*, *36*(6), 70–79.

Weatherford, L. R., & Kimes, S. E. (2003). A comparison of forecasting methods for hotel revenue management. *International journal of forecasting*, *19*(3), 401–415.

Williamson, E. L. (1992). *Airline network seat inventory control: Methodologies and revenue impacts* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Yao, X. (2002). Evolving artificial neural networks. *Proceedings of the IEEE*, *87*(9), 1423–1447.