On the role of (machine) learning in (mathematical) optimization

Andrea Lodi

Canada Excellence Research Chair
École Polytechnique de Montréal, Québec, Canada
andrea.lodi@polymtl.ca

CPAIOR 2017
@ Padova, (Italy), June 6, 2017

CANADA
EXCELLENCE
RESEARCH
CHAIR

DATA SCIENCE
FOR REAL-TIME
DECISION-MAKING

Outline

1. Two of my favorite examples of Big Data

2. Something I do find interesting in Big Data:
    1. New (business) models
    2. Formulating and solving integrated models

3. The role of learning:
    1. An example in Retail
    2. Machine Learning paradigm

4. Machine Learning and Mathematical Optimization:

    Q1: What can (Integer) Optimization do for Machine Learning?

    Q2: **What can Machine Learning do for Optimization?**

    Q3: What's new by the combination of Learning and Optimization?

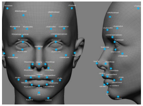
5. Conclusions

Ex. 1: automatic data collection (aka nowhere to hide)

A face recognition system has been put in place in a mall somewhere in the US.

Main purpose of the system was security.

After collecting data for some time, it has been observed that the large majority of the clients entering in the mall around lunch time (11 AM - 3 PM) was composed by Asian-American people.

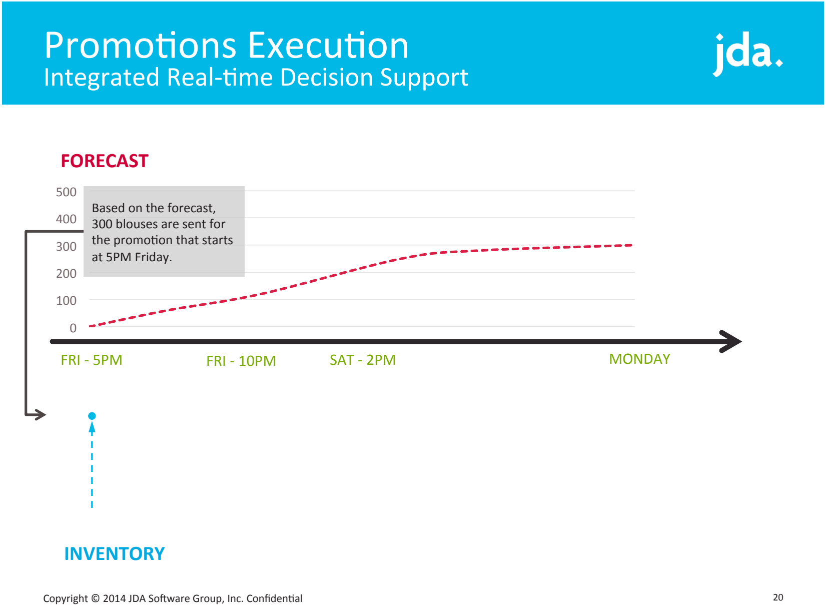


The company owning the mall implemented two simple actions:

- revised the shifts of the employees so as that (most of) the Asian-American ones were on duty in that time window;
- hired new Asian-American employees.

The overall effect has been a huge increase in sales.

Ex. 2: integrated decision support



Promotions Execution

Integrated Real-time Decision Support

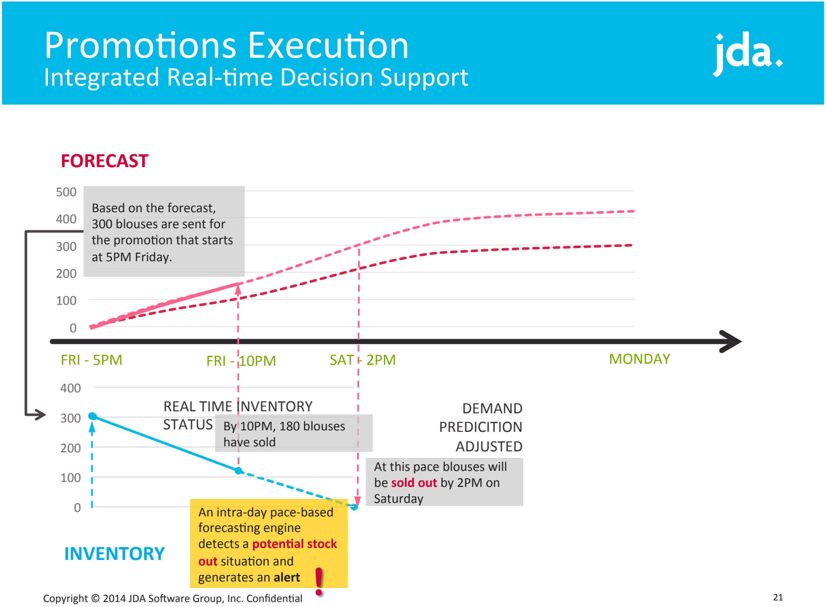
jda.

FORECAST

Based on the forecast, 300 blouses are sent for the promotion that starts at 5PM Friday.

INVENTORY

Ex. 2: integrated decision support



Promotions Execution

Integrated Real-time Decision Support

jda.

FORECAST

Based on the forecast, 300 blouses are sent for the promotion that starts at 5PM Friday.

REAL TIME INVENTORY STATUS

By 10PM, 180 blouses have sold

DEMAND PREDICITION ADJUSTED

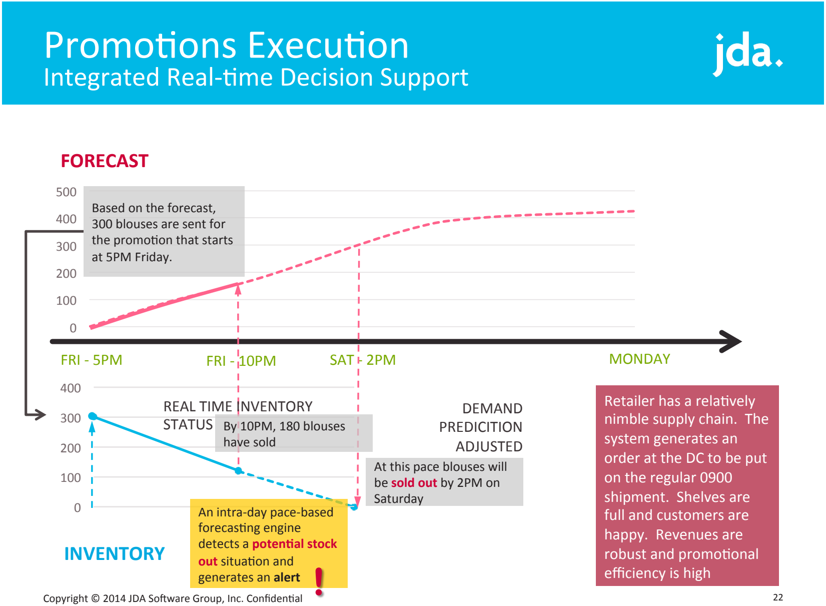
At this pace blouses will be **sold out** by 2PM on Saturday

INVENTORY

An intra-day pace-based forecasting engine detects a **potential stock out** situation and generates an **alert** !

21

Ex. 2: integrated decision support



Promotions Execution

Integrated Real-time Decision Support

Based on the forecast, 300 blouses are sent for the promotion that starts at 5PM Friday.

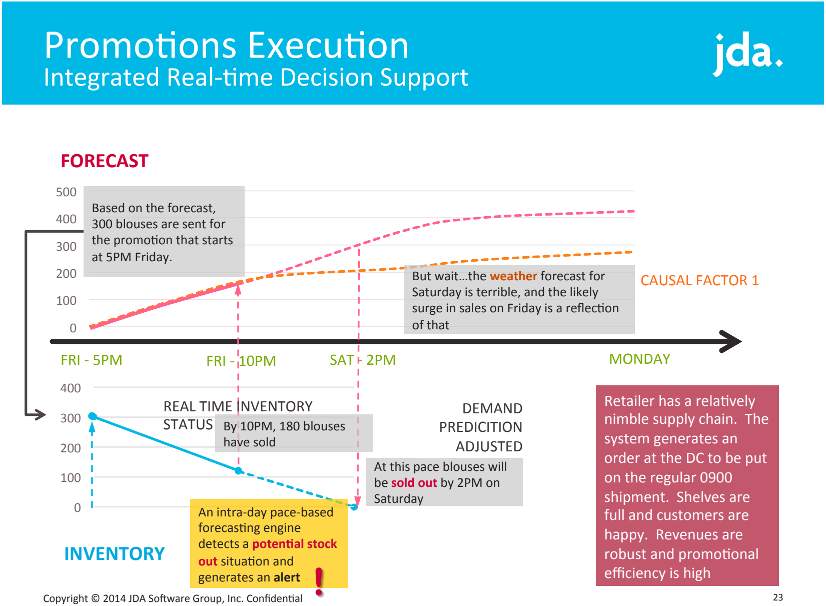
By 10PM, 180 blouses have sold

At this pace blouses will be **sold out** by 2PM on Saturday

An intra-day pace-based forecasting engine detects a **potential stock out** situation and generates an **alert** !

Retailer has a relatively nimble supply chain. The system generates an order at the DC to be put on the regular 0900 shipment. Shelves are full and customers are happy. Revenues are robust and promotional efficiency is high

Ex. 2: integrated decision support



Promotions Execution

Integrated Real-time Decision Support

Based on the forecast, 300 blouses are sent for the promotion that starts at 5PM Friday.

But wait…the **weather** forecast for Saturday is terrible, and the likely surge in sales on Friday is a reflection of that

REAL TIME INVENTORY STATUS

By 10PM, 180 blouses have sold

DEMAND PREDICITION ADJUSTED

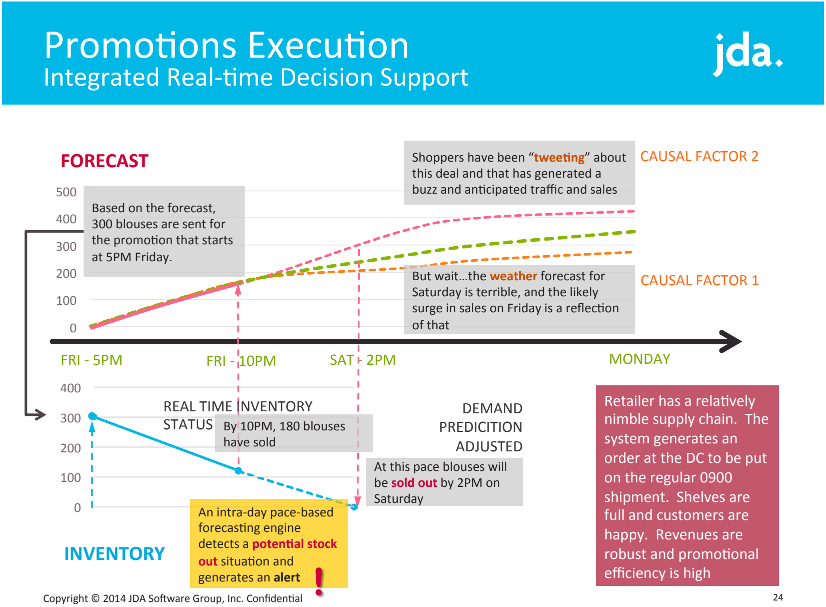
At this pace blouses will be **sold out** by 2PM on Saturday

An intra-day pace-based forecasting engine detects a **potential stock out** situation and generates an **alert** !

Retailer has a relatively nimble supply chain. The system generates an order at the DC to be put on the regular 0900 shipment. Shelves are full and customers are happy. Revenues are robust and promotional efficiency is high

23

Ex. 2: integrated decision support



Promotions Execution

Integrated Real-time Decision Support

jda.

Based on the forecast, 300 blouses are sent for the promotion that starts at 5PM Friday.

Shoppers have been “**tweeting**” about this deal and that has generated a buzz and anticipated traffic and sales

CAUSAL FACTOR 2

But wait…the **weather** forecast for Saturday is terrible, and the likely surge in sales on Friday is a reflection of that

CAUSAL FACTOR 1

REAL TIME INVENTORY STATUS

By 10PM, 180 blouses have sold

DEMAND PREDICITION ADJUSTED

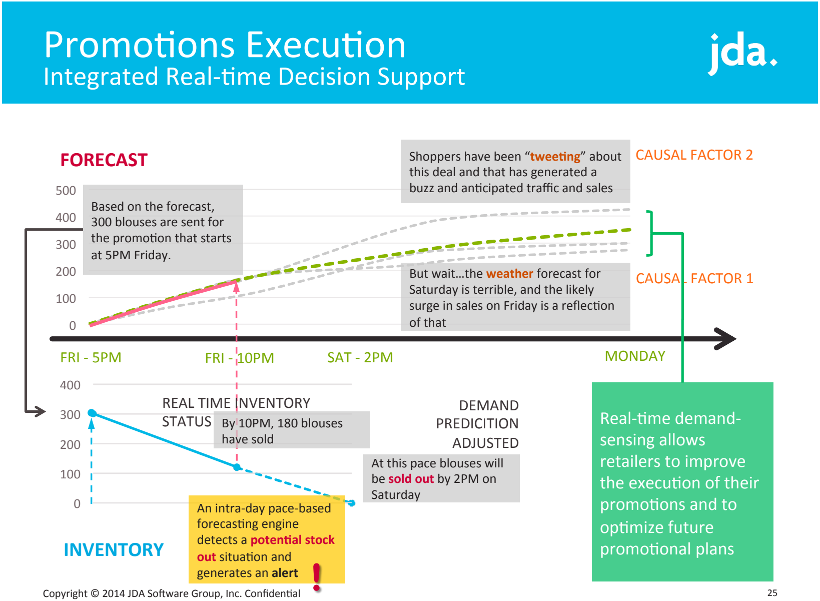
At this pace blouses will be **sold out** by 2PM on Saturday

An intra-day pace-based forecasting engine detects a **potential stock out** situation and generates an **alert** !

Retailer has a relatively nimble supply chain. The system generates an order at the DC to be put on the regular 0900 shipment. Shelves are full and customers are happy. Revenues are robust and promotional efficiency is high

24

Ex. 2: integrated decision support



Promotions Execution
Integrated Real-time Decision Support

jda.

FORECAST

500
400
300
200
100
0

Based on the forecast, 300 blouses are sent for the promotion that starts at 5PM Friday.

Shoppers have been "**tweeting**" about this deal and that has generated a buzz and anticipated traffic and sales

CAUSAL FACTOR 2

But wait…the **weather** forecast for Saturday is terrible, and the likely surge in sales on Friday is a reflection of that

CAUSAL FACTOR 1

FRI - 5PM
FRI - 10PM
SAT - 2PM
MONDAY

400
300
200
100
0

REAL TIME INVENTORY STATUS

By 10PM, 180 blouses have sold

DEMAND PREDICITION ADJUSTED

At this pace blouses will be **sold out** by 2PM on Saturday

An intra-day pace-based forecasting engine detects a **potential stock out** situation and generates an **alert** !

INVENTORY

Real-time demand-sensing allows retailers to improve the execution of their promotions and to optimize future promotional plans

25

[Côté, 2015]

What I like of Big Data

The first example shows that automatic collection of data can lead to the definition of new (optimization) problems.

Disseminating sensors (including mobile devices) everywhere has become cheap (and cool!) but the real challenge is taking decisions over the collected (complex) data.

It is not completely clear if the (applied) optimization problems we were used to solve in contexts as diverse as routing, supply chain and logistics, energy, telecommunications, etc. are still there or, instead, have radically changed.

The spirit of such a change is shown by the second example: the end-users behavior/preference is putting more and more pressure on the decision makers and, by transitivity, on the optimizers. This is not true only in the retail industry but virtually in any other in which a service is delivered:

- routing, I can check with my mobile device where cabs/buses are located;
- traffic management, I am aware of congestions, accidents, etc. in the city;
- cache allocation for video streaming, complaints escalate in real time.

What I like of Big Data (cont.d)

The most significant effect of considering the end-users behavior is that complex systems that have been traditionally split into smaller parts, optimized sequentially, now need to be tackled in an integrated fashion. Splitting was happening because of

1. difficulty and cost of collecting reliable data for the entire system
2. the size of the decision problems associated with considering the entire system would have been too large
3. there was very little perception both among the industrial players and among the end-users that splitting was avoidable.

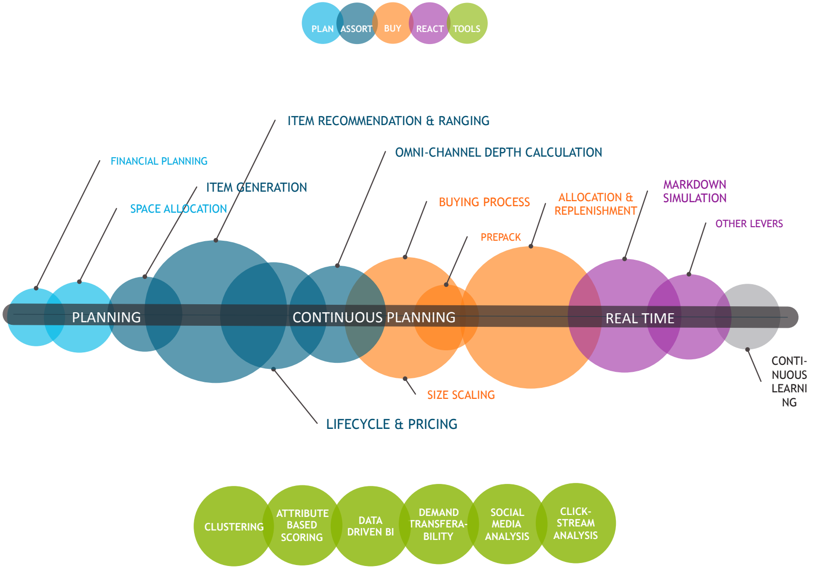
Lack of technological communication:

- the different divisions of, say, a firm, had little data exchange, and
- the end-user had no mobile technology to be updated in real time.

Mobile technology has urged the request of integrated approaches for decision making because of the perception of missing opportunities.

This is true in energy as well, where smart meters and smart buildings (producing energy as well as consuming it) are increasing end-users' awareness and pushing for more (integrated) optimization.

Integrated models: (the dream of) big data in retail



[6] [Côté, 2015]

The role of learning

From an optimization perspective, formulating and solving those integrated models is, of course, hard.

This is because of

1. volume
2. velocity
3. variety

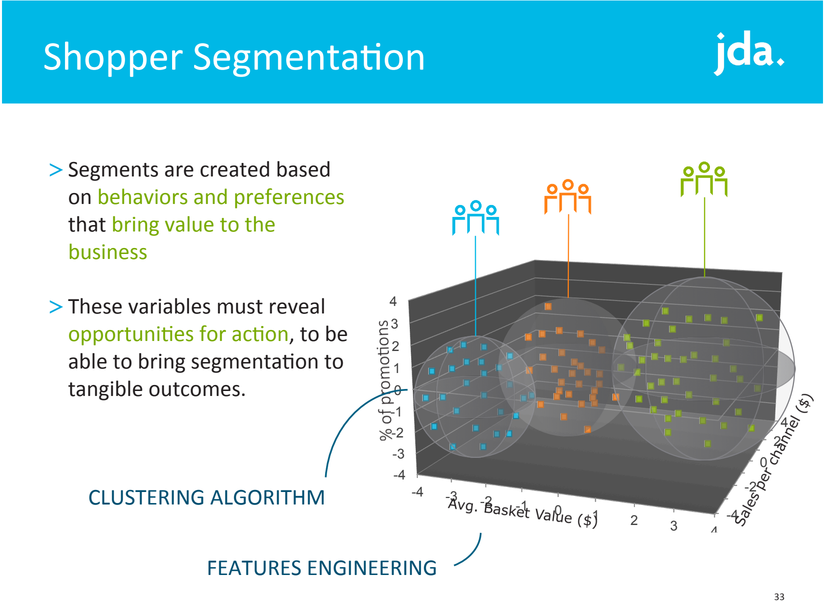
of the data, and also because optimizers are not – in general – trained for that.

One answer to this is introducing into the picture some learning mechanisms that allow to treat data, often reducing their volume and variety, and to take into account the end-user perspective/behavior.

In the retail context, one needs to predict the sales of a certain product, on a certain shop location, in a certain season, to a certain segment of shoppers.

Learning from historical data allows to compute a score associated with these choices and the optimization problem associated with the assortment can be solved only after these scores are computed.

Ex. 3: taking into account the end-user



Shopper Segmentation

jda.

- Segments are created based on behaviors and preferences that bring value to the business
- These variables must reveal opportunities for action, to be able to bring segmentation to tangible outcomes.

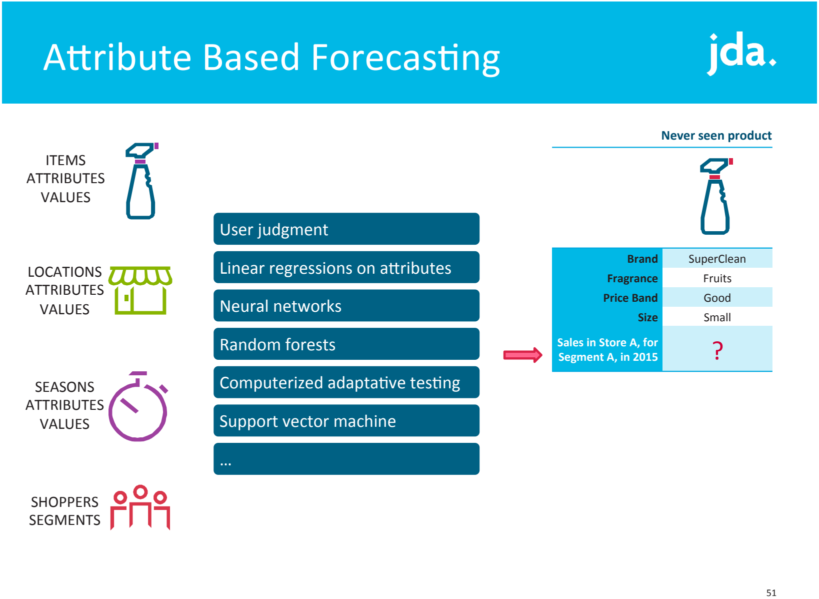
CLUSTERING ALGORITHM

FEATURES ENGINEERING

33

[Côté (2015)]

Ex. 3: taking into account the end-user (cont.d)



Attribute Based Forecasting

jda.

ITEMS
ATTRIBUTES
VALUES

LOCATIONS
ATTRIBUTES
VALUES

SEASONS
ATTRIBUTES
VALUES

SHOPPERS
SEGMENTS

- User judgment
- Linear regressions on attributes
- Neural networks
- Random forests
- Computerized adaptative testing
- Support vector machine
- ...

Never seen product

Brand	SuperClean
Fragrance	Fruits
Price Band	Good
Size	Small
Sales in Store A, for Segment A, in 2015	?

51

[Côté (2015)]

Machine Learning

Generally speaking, Machine Learning is a collection of techniques for

learning patterns in or

understanding the structure of **data**,

often with the aim of performing data mining, i.e., recovering previously unknown, actionable information from the learnt data.

Typically, in ML one has to “learn” from data (points in the so-called training set) a (nonlinear) function that predicts a certain score for new data points that are not in the training set.

Each data point is represented by a set of features, which define its characteristics, and whose patterns should be learnt.

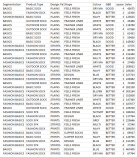
The techniques used in ML are diverse, going from artificial neural networks, to first order methods like gradient descent, to convex optimization, etc.

Machine Learning in retail

Learning Process

jda.

Historical data

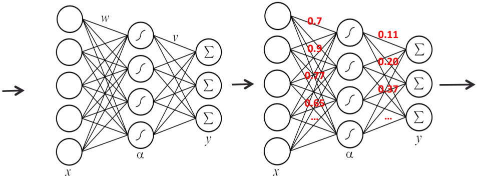


Segmentation	Product.Type	Design.Pa	Shape	Colour	GBB	space	Sales
BASICS	BASIC SOCK	PLAINS	FEELS FRESH	GRY MA	GOOD	4	45675
BASICS	FASHION SOCK	PLAINS	FEELS FRESH	BLACK	BETTER	1	13300
FASHION BASICS	BASIC SOCK	PLAINS	FEELS FRESH	NAVY	BETTER	3	125625
BASICS	OUTDOOR SOCK	PLAINS	TRAINER LINER	WHITE	BETTER	3	25466
BASICS	OUTDOOR SOCK	PLAINS	TRAINER LINER	WHITE	GOOD	1	116760
BASICS	OUTDOOR SOCK	PLAINS	TRAINER LINER	WHITE	GOOD	4	67368
BASICS	BASIC SOCK	PLAINS	FEELS FRESH	GRY MA	GOOD	1	61432
BASICS	BASIC SOCK	PLAINS	FEELS FRESH	GRY MA	GOOD	3	50456
BASICS	BASIC SOCK	PLAINS	FEELS FRESH	BLACK	GOOD	2	145662
FASHION BASICS	FASHION SOCK	STRIPES	FEELS FRESH	BLACK	BETTER	2	17375
FASHION BASICS	FASHION SOCK	STRIPES	FEELS FRESH	MULTI	BETTER	4	45765
BASICS	OUTDOOR SOCK	PLAINS	TRAINER LINER	WHITE	BETTER	4	212340
FASHION BASICS	FASHION SOCK	STRIPES	FEELS FRESH	MULTI	BETTER	4	356220
FASHION BASICS	FASHION SOCK	STRIPES	FEELS FRESH	GRY MA	BETTER	4	27408
BASICS	OUTDOOR SOCK	PLAINS	TRAINER LINER	WHITE	GOOD	1	209832
FASHION BASICS	BASIC SOCK	PLAINS	FEELS FRESH	NAVY	BETTER	2	189245
FASHION BASICS	FASHION SOCK	STRIPES	DESIGN	BLUE	BETTER	3	327723
FASHION BASICS	FASHION SOCK	STRIPES	FEELS FRESH	GRY MA	BETTER	3	220482
BASICS	OUTDOOR SOCK	PLAINS	TRAINER LINER	WHITE	GOOD	1	71100
FASHION BASICS	BASIC SOCK	PLAINS	FEELS FRESH	BLACK	BETTER	4	282100
FASHION BASICS	FASHION SOCK	PLAINS	FEELS FRESH	BLU/BK	BETTER	4	305164
FASHION BASICS	FASHION SOCK	PLAINS	FEELS FRESH	GRY MA	BETTER	1	352000
FASHION BASICS	OUTDOOR SOCK	STRIPES	TRAINER LINER	WHITE	BETTER	2	147360
FASHION BASICS	BASIC SOCK	PLAINS	FEELS FRESH	BLACK	BETTER	3	187077
BASICS	OUTDOOR SOCK	PLAINS	TRAINER LINER	WHITE	GOOD	3	229236
FASHION BASICS	SOCK 5PK	PRINTS	FEELS FRESH	NVYGRY	BETTER	1	180224
FASHION BASICS	FASHION SOCK	PRINTS	DESIGN	BLACK	BETTER	3	137764
FASHION BASICS	SOCK 5PK	PRINTS	FEELS FRESH	GREY	BETTER	3	225666
FASHION BASICS	SOCK 5PK	STRIPES	FEELS FRESH	MULTI	BETTER	4	37248
BASICS	FASHION SOCK	PRINTS	DESIGN	NAVY	BETTER	4	284050
FASHION BASICS	BASIC SOCK	PRINTS	SLIPPER SOCKS	RED	BETTER	1	69927
FASHION BASICS	BASIC SOCK	PRINTS	SLIPPER SOCKS	NAVY	BETTER	2	120736
FASHION BASICS	FASHION SOCK	PLAINS	DESIGN	BLUE	BEST	4	311395
FASHION BASICS	SOCK 5PK	STRIPES	FEELS FRESH	BLACK	BETTER	4	27384
BASICS	FASHION SOCK	PRINTS	DESIGN	BLUE	BETTER	3	467400
FASHION BASICS	FASHION SOCK	STRIPES	FEELS FRESH	GRY MA	BETTER	3	67784

Model Training

Trained Model

Score



$$y(x) = \sum_{j=1}^{m} v_i \sigma(\sum_{i=1}^{n} w_{ij} x_i)$$

Compute the

W and **V**

W and **V** are now known

69 [Côté (2015)]

ML & Mathematical Optimization

I believe big data applications call for the integration between Machine Learning and Mathematical Optimization.

But, how such an integration should go?
And, what about Mixed-Integer Programming (MIP) specifically?

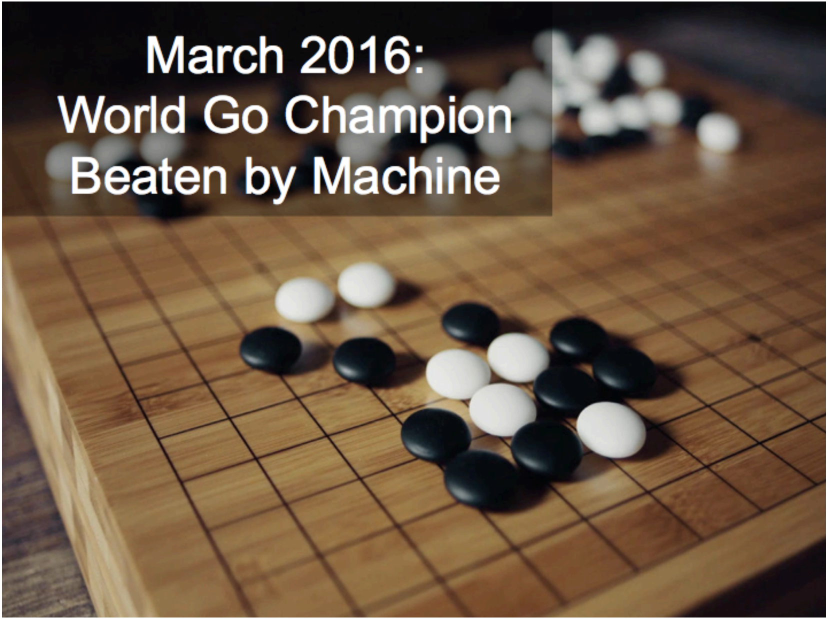
Of course, the easiest integration is already shown in the examples above, where raw data are "crunched" and "prepared" by Machine Learning to construct the decision model on which Mathematical Optimization is applied.

However, the integration is not restricted to let ML and MP work in cascade.

Modern ML paradigms like deep learning (essentially, learning by multiple layers) are facing more and more complicated structures in which the features (raw data observations) are not kept fixed but are "transformed" within the learning process.

Those transformations involve highly nonconvex functions and discrete decisions.

ML, Deep Learning and Reinforcement Learning



Q1: What can (integer) Optimization do for ML?

Discrete decisions have been disregarded so far in ML.

This is certainly due to the (negative) perception that were not affordable in practical computation (ML has always been concerned with large volumes of data) but it was also related to the fact that the parameters to be learnt were inherently continuous.

This might be less true in modern paradigms, those that led ML to contribute to the advances in computer vision, signal processing and speech recognition.

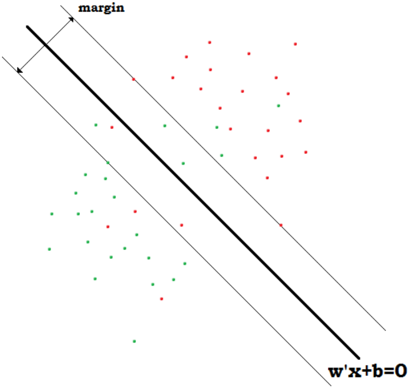
Moreover, there seems to be large room for using discrete variables to formulate nonconvexities that appear more and more to be crucial in ML.

Q1: Discrete decisions in Support Vector Machine

Ramp Loss

$$\min \frac{\omega^\top \omega}{2} + \frac{C}{n}\sum_{i=1}^{n} \xi_i$$

$$\begin{aligned} y_i(\omega^\top x_i + b) \geq 1 - \xi_i & \quad \forall i = 1, \dots, n \\ 0 \leq \xi_i & \quad \forall i = 1, \dots, n \\ \omega \in \mathbb{R}^d, b \in \mathbb{R} & \end{aligned}$$



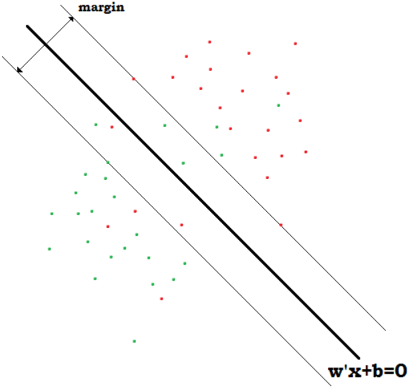
Q1: Discrete decisions in SVM (cont.d)

Ramp Loss $g(\xi_i) = (\min\{\xi_i, 2\})^+$

$$\min \frac{\omega^\top \omega}{2} + \frac{C}{n}\left(\sum_{i=1}^{n} \xi_i + 2\sum_{i=1}^{n} z_i\right)$$

$$\begin{aligned} y_i(\omega^\top x_i + b) &\geq 1 - \xi_i - M z_i \quad && \forall i = 1, \dots, n \\ 0 \leq \xi_i &\leq 2 && \forall i = 1, \dots, n \\ \omega \in \mathbb{R}^d&, b \in \mathbb{R} \\ z &\in \{0, 1\}^n \end{aligned}$$

with $M > 0$ big enough constant.



[Brooks (2011)]

Sophisticated methods for dealing with big-M constraints in MIP have been recently devised and integrated within the IBM-Cplex solver, so as decent-size SVM instances above can now be routinely solved to optimality.

[Belotti, Bonami, Fischetti, Lodi, Monaci, Nogales & Salvagnin (2016)]

Q1: What can (integer) Optimization do for ML? (cont.d)

The described one is just one example of a potential area of interaction:

1. there are ML problems that are naturally casted as MIPs,
2. there is a bunch of work to try to solve them **NOT** as MIPs,
3. maybe there is something to be gained in treating them as MIPs!

Other examples are in semi-supervised SVM and its multi-category generalizations. [Bennett & Demiriz (1998), Lodi & Pouliot (working paper)]

An additional area of interaction is the so-called hyper-parameter optimization, where the parameters of a (deep) neural network have to be optimized so as to make the learning effective. [Audet & Orban (2007, . . .)]

MIP (mostly, Combinatorial Optimization) sub-structure are present in Structured Prediction problems. Namely, these are the ML problems in which some constraints on the structure of the prediction have to be satisfied.

A classical example is word alignment (a key step in machine translation), where matching and transportation structures can be effectively exploited.

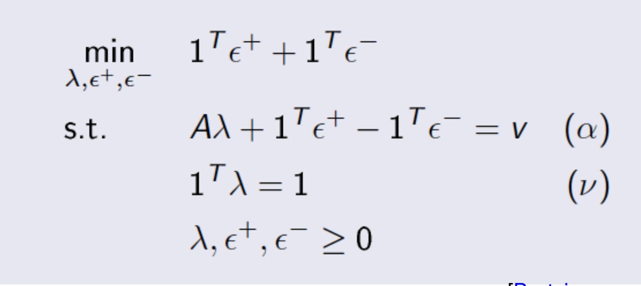
[Lacoste-Julien et al. (2006, 2013, . . .)]

Q1: Learning by Column Generation

Besides formulating learning / classification problems by IP, one can apply sophisticated IP techniques to the learning phase.

This is the case of training a choice model in assortment optimization, where given a subset of the consumer's behaviors, one has to find the probability distribution (λ_k) that explains at best the training set, i.e., the observed sales.

This can be done in a very elegant way by Column Generation



$$\begin{aligned} \min_{\lambda,\epsilon^+,\epsilon^-} \quad & 1^T\epsilon^+ + 1^T\epsilon^- & \\ \text{s.t.} \quad & A\lambda + 1^T\epsilon^+ - 1^T\epsilon^- = v & (\alpha) \\ & 1^T\lambda = 1 & (\nu) \\ & \lambda, \epsilon^+, \epsilon^- \geq 0 & \end{aligned}$$

[Bertsimas and Misic (2015)]

and the challenge is to make it practical for relevant sizes of the number of products. [Jena, Lodi and Palmer (2017)]

Q2: What can ML do for (Integer) Optimization?

A fast growing literature has started to appear in the last 5 to 10 years on the use of Machine Learning techniques to help Optimization, especially MIP solvers. Among the first in these series, the papers on tuning MIP solvers. [Hoos et al. (2010+)]

ML has, of course, started to be used within Constraint Programming as well, including Neural Networks and Decision Trees. [Lombardi & Milano (2015+)]

Learning when to use a decomposition. [Kruber, Lübbecke, Parmentier (2017): tomorrow!]

MIP solvers are complex software objects implementing a large variety of algorithmic approaches. Strategic decisions on how to combine those approaches in the most effective way have to be taken over and over. Such decisions are taken heuristically, often breaking ties in architecture-dependent ways, thus showing the heuristic nature of MIP implementations. [Lodi (2012)]

ML can help systematize the process that leads to take these decisions, especially when a large quantity of data can be collected.

Q2: Variable selection in Branch and Bound

Branch-and-Bound algorithm (B&B):

- most widely used procedure for solving (Mixed-)Integer Programming problems
- implicit enumeration search, mapped into a decision tree
- leave (at least) two big choices:
  1. How to split a problem into subproblems (variable selection)
  2. Which node/subproblem to select for the next exploration

. . . decisions play a key role for the algorithm efficiency!

- as of today, decisions are made heuristically and empirically evaluated
- there are good branching strategies, but usually very costly

Ultimate goal (details to be discussed in slot #4)

Use ML to learn an activation function that can be adopted as approximation / prediction of a good B&B strategy, ideally with a low computational cost.

[Alvarez, Wehenkel & Louveaux (2016), Khalil, Le Bodic, Song, Nemhauser & Dilkina (2016)]

Q2: MIQPs classification

We consider Mixed-Integer Quadratic Programming (MIQP)

$$\begin{aligned} \min \quad & \frac{1}{2}x^TQx + c^Tx \\ & Ax = b \\ & x_i \in \{0,1\} \quad \forall\, i \in I \\ & l \leq x \leq u \end{aligned} \tag{1}$$

where $Q = \{q_{ij}\}_{i,j=1\dots n} \in \mathbb{R}^{n\times n}$ is a real symmetric matrix, either convex or nonconvex, and all integer variables are binary.

Depending on the problems' structure, we can tackle them in different ways:

- $Q \succeq 0$: perform NLP based B&B (or Outer Approximation algorithms)
- $Q \not\succeq 0$: depending on variables' type,
    - pure 0-1: transform into either a convex MIQP or into a MILP (i.e., linearize it)
    - mixed: perform $Q-$space reformulation/relaxation, run Global Optimization algorithms (Spatial B&B)

Q2: MIQPs classification (cont.d)

The linearization approach seems beneficial also for the convex case, both for pure 0-1 and mixed problems. However, is linearizing always the best choice?

"[. . .] when one looks at a broader variety of test problems the decision to linearize (vs. not linearize) does not appear so clear-cut.[1]*"*

Exploit ML predictive machinery to understand whether it is favorable to linearize the quadratic part of the MIQP or not

- Learn an offline classifier predicting the most suited resolution approach within `IBM-CPLEX` framework (`qtolin` linearization switch parameter)
- Gain theoretical insights about which features of the MIQPs most affect the prediction

[Bonami, Lodi, Zarpellon (working paper)]

[1] Fourer B. Quadratic Optimization Mysteries, Part 2: Two Formulations. `http://bob4er.blogspot.com/2015/03/quadratic-optimization-mysteries-part-2.html`

Q2: MIQPs classification - Dataset generation

. . . traditional benchmark sets are too small for learning!

We define and implement a generator of MIQP instances, spanning a variety of structural parameters and optimization components.

(I) Objective function data generation: real symmetric matrices are generated via the MATLAB function

```
Q = sprandsym(size, density, eigenvalues)
```

(II) Variables' type definition: binary / continuous variables are added to the problems with respect to the sign of Q, in different proportions

(III) Constraints generation: different constraints sets are added accordingly to the type of variables of the problem (e.g., cardinality, simplex, multi-dimensional knapsack)

- Dataset of 2300 instances, three types of MIQPs (0-1 convex, 0-1 nonconvex, mixed convex)
- Plan to compare with traditional benchmark libraries for test / extensions

Q2: MIQPs classification - Features design

We define a set of 23 features referring to an MIQP instance, and we divide them into two main blocks:

- Static features describe the mathematical characteristics of the instance, in terms of
  - variables - e.g., number, types, presence in constraints and objective
  - constraints - e.g., coefficients and variables presence
  - quadratic objective function - e.g., coefficients, variables presence, sparsity, spectral properties

  They are extracted via `CPLEX/Python` before any solving (pre)process takes place.

- Dynamic features describe the initial behavior with respect to different resolution methods.
  - e.g., bounds and solution times at the root node

  They are extracted from the early stages of the optimization, after the preprocessing and the resolution of the root node relaxation.

Q2: MIQPs classification - Labeling procedure

One of three different labels among $\{\mathrm{L}, \mathrm{NL}, \mathrm{T}\}$ can be assigned to a MIQP instance, describing the winner between *linearize*, *not-linearize* or the case of a *tie* of the two methods.

Each problem of the dataset is run with timelimit of 1h, for 5 different random seeds, with `qtolin` on and off.

To address solvability / consistency issues, we perform

- Solvability check, to discard never-solved instances
- Seed consistency check on each seed, to discard unstable instances w.r.t. the found upper and lower bounds
- Global consistency check on global best upper and lower bounds, to discard unstable instances

Running time is the ultimate compared measure, assessing the final label for each example.

Q2: MIQPs classification - Learning experiments

Instances, features and labels give a dataset ready for supervised learning:

$$\{(x^i, y^i)\}_{i=1..N} \quad \text{where } x^i \in \mathbb{R}^d,\ y \in \{\mathrm{L}, \mathrm{NL}, \mathrm{T}\} \quad \text{for } N \text{ MIQPs}$$

Multiclass classifiers such as

- Support Vector Machines (nonlinear RBF kernel) (SVM)

and ensemble methods based on Decision Trees (more interpretable than Neural Networks), such as

- Random Forests (RF)
- Extremely Randomized Trees (EXT)
- Gradient Tree Boosting (GTB)

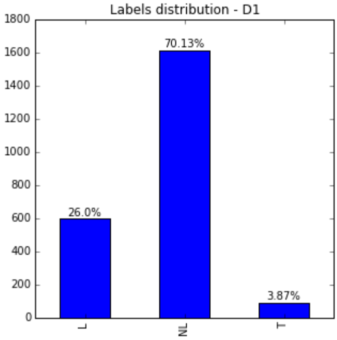
Methodology: follow ML best practices to avoid overfitting

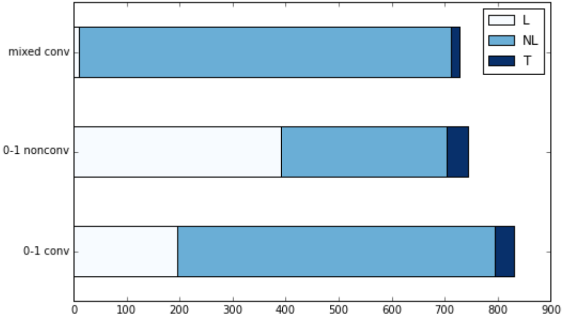
- Training phase to optimize parameters (1725 instances)
- *k*-fold cross validation and grid search for hyper-parameters selection
- Test phase to assess classifiers' performance (575 instances)

Main implementation tool: `scikit-learn` library.

Q2: MIQPs classification - Nutshell analysis

Before learning, look into the dataset! In a nutshell:





- Take care of unbalanced data in the learning procedure
- Can some trends already been recognized w.r.t. different problem types?
- More (statistical) analyses on features and data distribution

Q2: MIQPs classification - Some results

Classifiers perform well with respect to traditional classification measures

	SVM	RF	EXT	GTB
Accuracy	0.85	0.89	0.85	0.87
Precision	0.82	0.87	0.82	0.85
Recall	0.85	0.89	0.85	0.87
F1 score	0.83	0.87	0.84	0.86

Are our predictors leading to a sensible gain in running times?

- For each example, depending on the predicted label, select the predicted mode (L, NL, T) running time. Take the shifted geometric mean of the prescribed times and normalize it w.r.t. the a posteriori best and worst labeling, to get a timing score $\in [0, 1]$

	RF	`CPLEX default`
Timing score	0.99	0.48

Q2: MIQPs classification - Going further

Deepen the features analysis:

- At a glance, dynamic features and static ones about eigenvalues are very important for the predictions

What about other datasets?

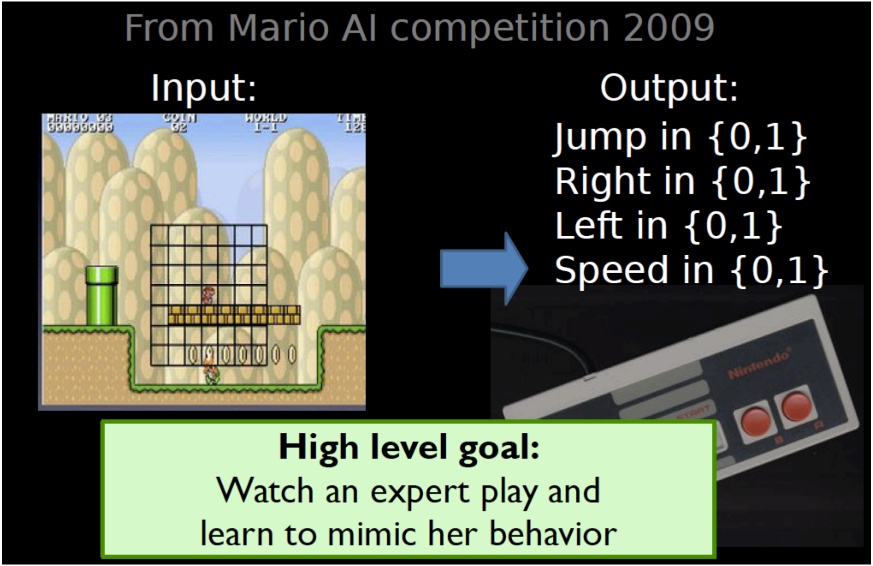
- First attempt on (part of) CPLEX internal testbed as new test set
- All classifiers perform poorly in terms of classification measures, but not so bad in terms of timing score

	RF	EXT	CPLEX default
Timing score	0.89	0.91	0.96

- Why? Very different distribution of features, problems types and labels w.r.t. the synthetic dataset, on which the predictors were trained

. . . there is much more to understand and test!

Q2: Learning to Search



From Mario AI competition 2009

Input:

Output:

Jump in {0,1}
Right in {0,1}
Left in {0,1}
Speed in {0,1}

High level goal:
Watch an expert play and
learn to mimic her behavior

[Langford and Daumé III, 2015]

Q3: The power of ML & Optimization together

Why are learning and optimization two faces of the same coin?

A nice example comes in healthcare, for the so-called personalized medicine.

ML could be used to predict the medical outcome that would follow from a particular choice of combination and dosage of different drugs and treatments for a patient over the course of a few months to come.

However, there could be an exponential number of such combinations to consider, and constraints to be satisfied (for example, because of known side-effects and resources).

Exhaustively searching in the space of such combinations is and will always be unpractical and mathematical optimization is likely to be the answer.

A new methodology integrating learning and optimization is required, and such a methodology is likely to be useful every time the space of predictions faces combinatorial explosion.

Conclusions

We have discussed a few important issues arising in big data (optimization), namely

- the change of perspective associated with dealing with the end-users behavior,
- the need of formulating and solving integrated models, and
- the role of (machine) learning.

I am an optimistic person, so I see huge opportunities through the interaction between Machine Learning and Mathematical Optimization, including / especially on the Integer Programming side.

There is plenty of room for contributing to the subject and . . .

. . . getting on board in Montréal!

CANADA EXCELLENCE RESEARCH CHAIR
DATA SCIENCE FOR REAL-TIME DECISION-MAKING
MILA
IVADO
INSTITUTE FOR DATA VALORISATION