On the interplay between Discrete Optimization and Machine Learning

Andrea Lodi

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

NGB/LNMB 2020 Seminar
"Optimization for and with Machine Learning"
@ Lunteren (The Netherlands), January 15, 2020

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)

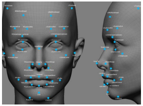
A face recognition system has been put in place in a mall somewhere in the US.

Main purpose of the system was security.

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.

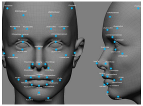


After collecting data for some time, it has been observed that the large majority of the clients entering in the mall around lunch time (11 AM - 3 PM) was composed by Asian-American people.

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.

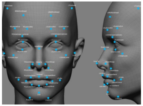
The company owning the mall implemented two simple actions:

- 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.

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

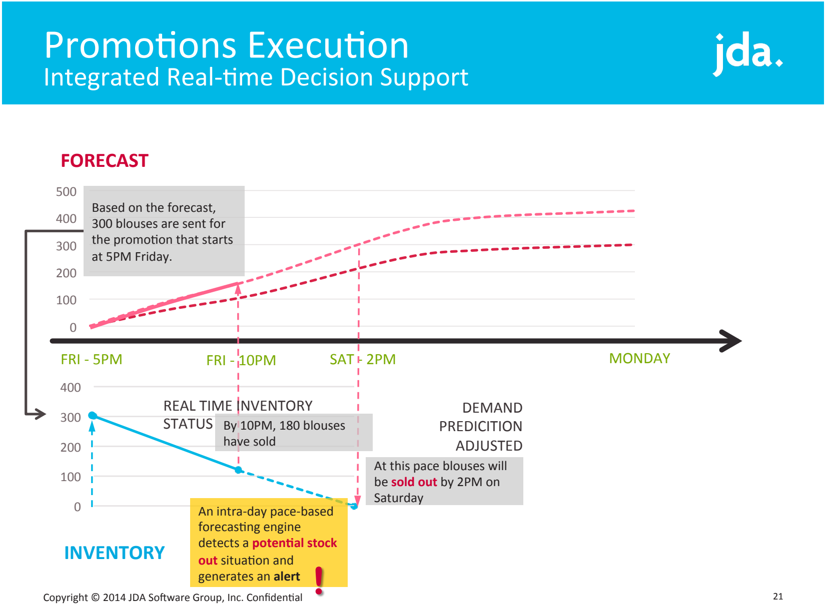
Promotions Execution
Integrated Real-time Decision Support

jda.

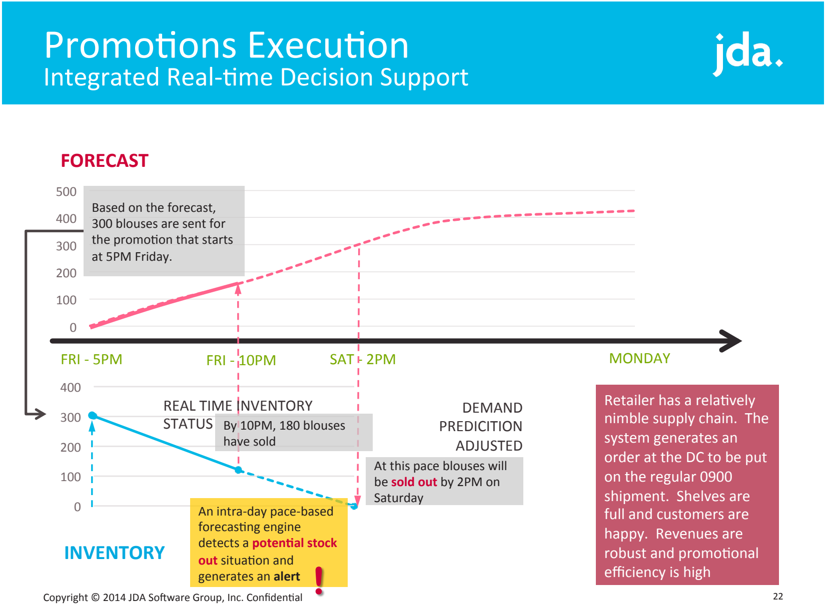
FORECAST

INVENTORY

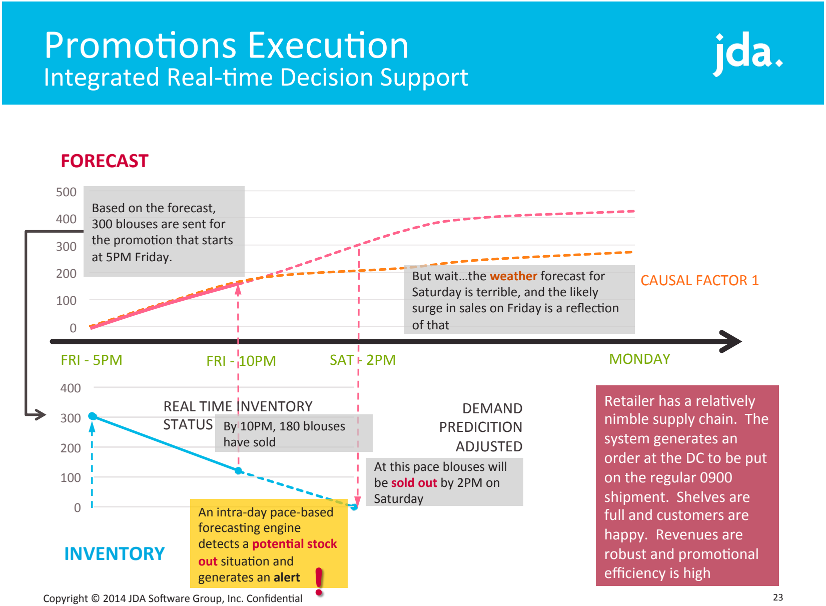
Ex. 2: integrated decision support



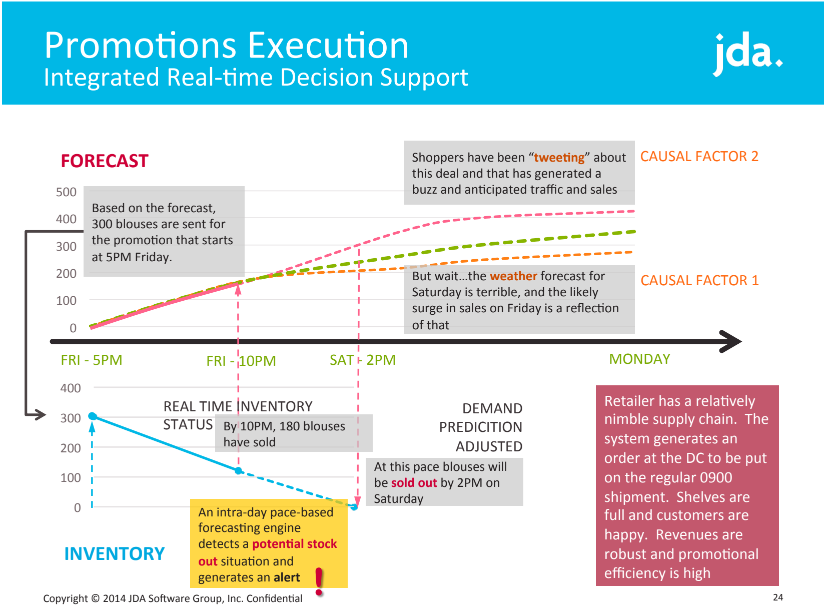
Ex. 2: integrated decision support



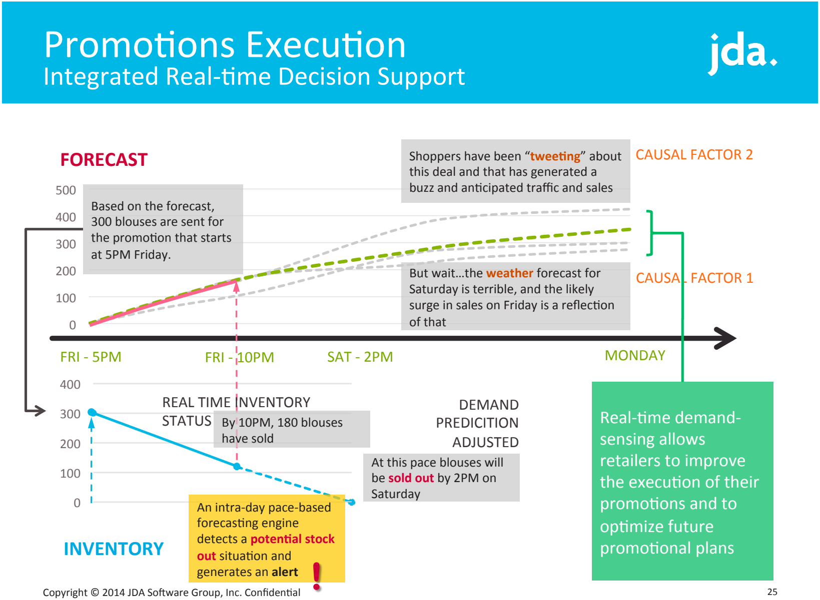
Ex. 2: integrated decision support



Ex. 2: integrated decision support



Ex. 2: integrated decision support



[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.

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.

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.

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.

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.

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.

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.

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.

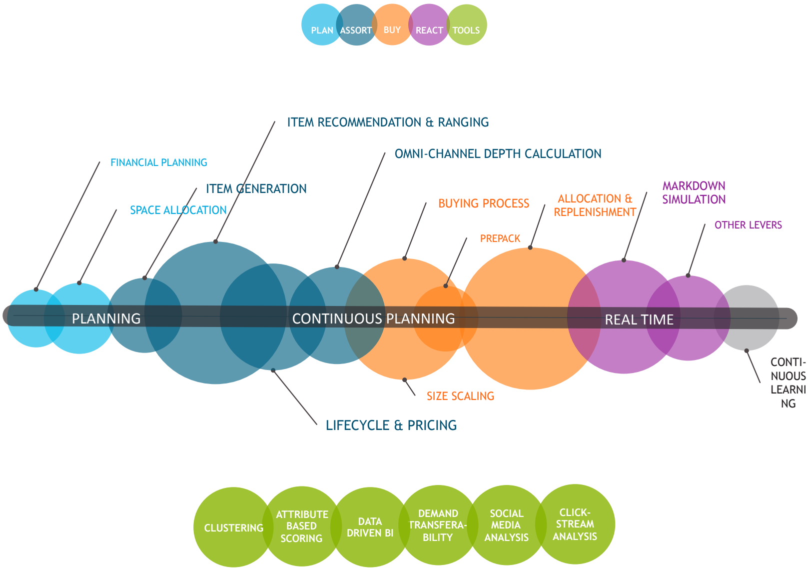
Lack of technological communication:

- 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.

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.

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.

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.

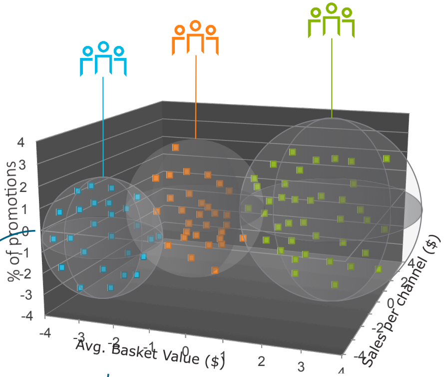
Learning from historical data allows to compute a score associated with these choices and the optimization problem associated with the assortment can be solved only after these scores are computed.

Ex. 3: taking into account the end-user

Shopper Segmentation

> 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.

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.

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.

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.

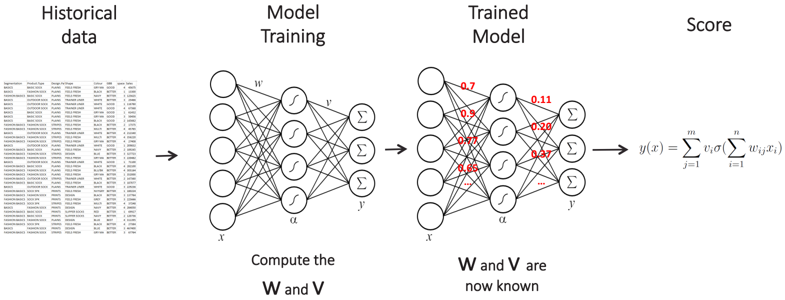
Typically, in ML one has to "learn" from data (points in the so-called training set) a (nonlinear) function that predicts a certain score for new data points that are not in the training set.

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.

Learning Process

jda.



[Côté (2015)]

ML & Mathematical Optimization

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

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?

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.

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.

On the other hand, instead on integrating them, an alternative would be to see ML and M(I)P in competition and, recently, there has been quite some work on both ways of this competition, i.e., by using MIP for solving classification problems and, vice versa, using ML to attack discrete optimization problems.

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?

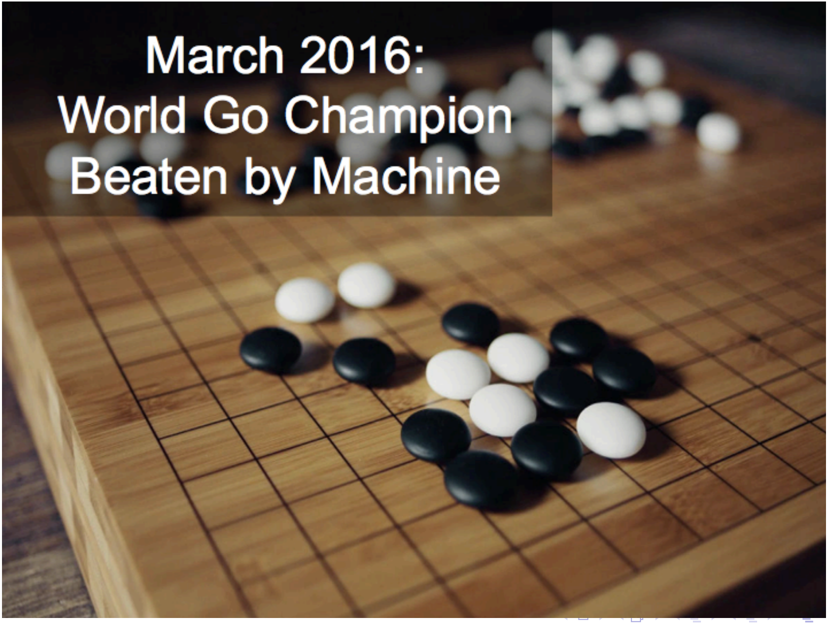
Of course, the easiest integration is already shown in the examples above, where raw data are "crunched" and "prepared" by Machine Learning to construct the decision model on which Mathematical Optimization is applied.

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

On the other hand, instead on integrating them, an alternative would be to see ML and M(I)P in competition and, recently, there has been quite some work on both ways of this competition, i.e., by using MIP for solving classification problems and, vice versa, using ML to attack discrete optimization problems.

We will briefly discuss those attempts but we will mostly concentrate on the integration.

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.

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.

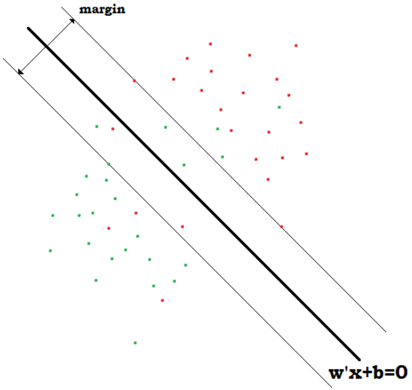
Moreover, there seems to be large room for using discrete variables to formulate nonconvexities that appear more and more to be crucial in ML.

Q1: Discrete decisions in Support Vector Machine

Ramp Loss

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

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



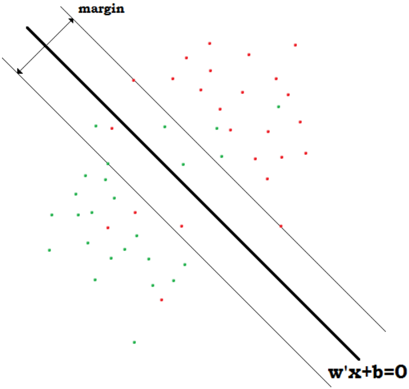
Q1: Discrete decisions in SVM (cont.d)

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

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

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

with $M > 0$ big enough constant.



[Brooks (2011)]

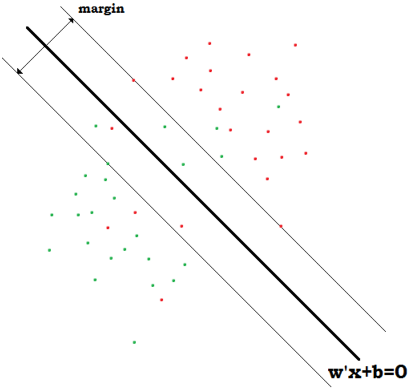
Q1: Discrete decisions in SVM (cont.d)

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

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

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

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)

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

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

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

An entire field of Interpretable Artificial Intelligence is emerging, where classification problems are solved by decision trees modeled as MIPs.

[Bertsimas et al. (2017), Günlük et al. (2018)]

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

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

An entire field of Interpretable Artificial Intelligence is emerging, where classification problems are solved by decision trees modeled as MIPs.

[Bertsimas et al. (2017), Günlük et al. (2018)]

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, . . .)]

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

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

An entire field of Interpretable Artificial Intelligence is emerging, where classification problems are solved by decision trees modeled as MIPs. [Bertsimas et al. (2017), Günlük et al. (2018)]

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.

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

Just one example: (1) there are ML problems that are naturally casted as MIPs (discrete), but (2) **NOT** solved as MIPs.

Here, the goal is not necessarily to use MIP only. However, leveraging the quality and experience of MIP solving for discrete problems can be a plus (bounds, rewards, interpretability, etc.)

An entire field of Interpretable Artificial Intelligence is emerging, where classification problems are solved by decision trees modeled as MIPs.

[Bertsimas et al. (2017), Günlük et al. (2018)]

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.

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

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

[Bertsimas and Misic (2015)]

and the challenge is to make it practical for relevant sizes of the number of products. [Jena, Lodi, Palmer, Sole (2017, 2019)]

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

A fast growing literature has started to appear in the last 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+)]

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

A fast growing literature has started to appear in the last 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 been used to devise constraints for a MIP, by modeling complex physical interactions: heating of processors in a data center.

[Lombardi & Milano (2015+)]

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

A fast growing literature has started to appear in the last 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 been used to devise constraints for a MIP, by modeling complex physical interactions: heating of processors in a data center.

[Lombardi & Milano (2015+)]

Learning when to use a decomposition.

[Kruber, Lübbecke, Parmentier (2017)]

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

A fast growing literature has started to appear in the last 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 been used to devise constraints for a MIP, by modeling complex physical interactions: heating of processors in a data center.

[Lombardi & Milano (2015+)]

Learning when to use a decomposition.

[Kruber, Lübbecke, Parmentier (2017)]

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.

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

A fast growing literature has started to appear in the last 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 been used to devise constraints for a MIP, by modeling complex physical interactions: heating of processors in a data center.

[Lombardi & Milano (2015+)]

Learning when to use a decomposition.

[Kruber, Lübbecke, Parmentier (2017)]

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)]

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

A fast growing literature has started to appear in the last 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 been used to devise constraints for a MIP, by modeling complex physical interactions: heating of processors in a data center.

[Lombardi & Milano (2015+)]

Learning when to use a decomposition.

[Kruber, Lübbecke, Parmentier (2017)]

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 MIPs
- 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

Q2: Variable selection in Branch and Bound

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

- most widely used procedure for solving MIPs
- 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
- good branching strategies are usually very costly (strong branching)

Q2: Variable selection in Branch and Bound

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

- most widely used procedure for solving MIPs
- 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
- good branching strategies are usually very costly (strong branching)

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: Variable selection in Branch and Bound

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

- most widely used procedure for solving MIPs
- 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
- good branching strategies are usually very costly (strong branching)

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)]

Most recent attempts include:

- Imitation learning with Graph Neural Networks (Monday) [Gasse et al. (2019)]
- Reinforcement learning [Zarpellon et al. (2020)]

Q2: selected topics

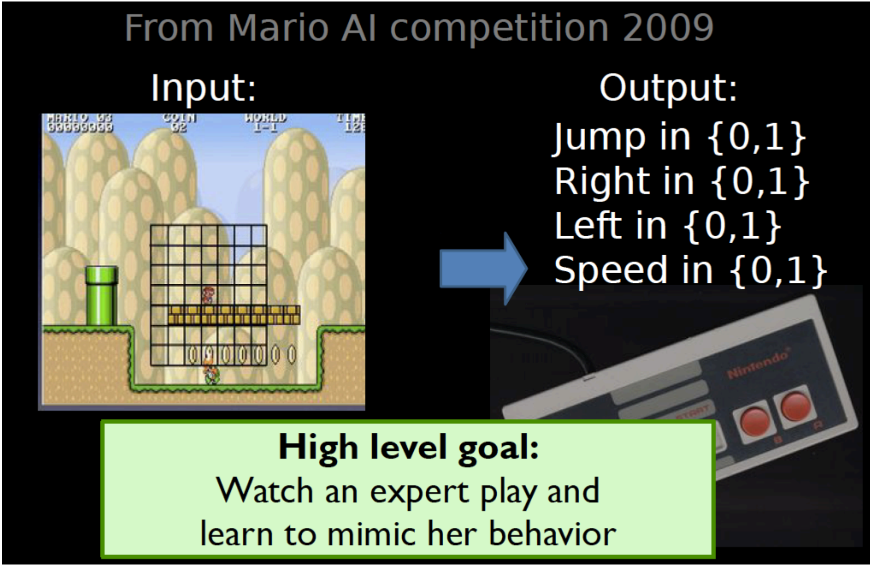
Learning MIQPs classification. [Bonami, Lodi & Zarpellon (2018)]

Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems. [Xavier, Qiu & Ahmed (2019)]

Learning Fast Optimizers for Contextual Stochastic Integer Programs. [Nair et al. (DeepMind) (2018)]

Predicting time limit / tree size for MIP resolution. [Zarpellon, Fischetti & Lodi (2019), Anderson et al. (2019)]

Q2: Learning to Search



[Langford and Daumé III, 2015]

Q2: Learning to Search (2)

The most notable outcome of the Learning to Search paradigm is the recent bulk of work on replacing MIP to solve combinatorial optimization problems by ML.

Q2: Learning to Search (2)

The most notable outcome of the Learning to Search paradigm is the recent bulk of work on replacing MIP to solve combinatorial optimization problems by ML.

Not surprising, the first attempts have been done in the Traveling Salesman Problem context and two papers stand:

- Supervised learning trained by precomputed (by an "expert") TSP solutions [Vinyals et al., 2015]
- Reinforcement learning with tour length as a reward function [Bello et al., 2017]

Q2: Learning to Search (2)

The most notable outcome of the Learning to Search paradigm is the recent bulk of work on replacing MIP to solve combinatorial optimization problems by ML.

Not surprising, the first attempts have been done in the Traveling Salesman Problem context and two papers stand:

- Supervised learning trained by precomputed (by an "expert") TSP solutions [Vinyals et al., 2015]
- Reinforcement learning with tour length as a reward function [Bello et al., 2017]

Currently, none of the approaches is competitive in any way with specifically designed algorithm but the research, admittedly, led to interesting ML architectures (that can be applied elsewhere).

Q2: Learning to Search (2)

The most notable outcome of the Learning to Search paradigm is the recent bulk of work on replacing MIP to solve combinatorial optimization problems by ML.

Not surprising, the first attempts have been done in the Traveling Salesman Problem context and two papers stand:

- Supervised learning trained by precomputed (by an "expert") TSP solutions [Vinyals et al., 2015]
- Reinforcement learning with tour length as a reward function [Bello et al., 2017]

Currently, none of the approaches is competitive in any way with specifically designed algorithm but the research, admittedly, led to interesting ML architectures (that can be applied elsewhere).

A different use of the same framework has been discussed yesterday afternoon in the context of tactical planning problems. [Larsen et al., 2019]

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.

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.

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.

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.

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!