Optimization for Machine Learning
Approximation Theory and Machine Learning

Sven Leyffer

Argonne National Laboratory

September, 30 2018
Outline

1. Data Analysis at DOE Light Sources
2. Optimization for Machine Learning
3. Mixed-Integer Nonlinear Optimization
   - Optimal Symbolic Regression
   - Deep Neural Nets as MIPs
   - Sparse Support-Vector Machines
4. Robust Optimization
   - Robust Optimization for SVMs
5. Conclusions and Extension
Motivation: Datanami from DOE Lightsource Upgrades

Data size and speed to outpace Moore’s law (source Ian Foster)

Light sources: 18 orders of magnitude in 5 decades

Computers: 12 orders of magnitude in 6 decades
Challenges at DOE Lightsources

Math, Stats, and CS Challenges from APS Upgrade

- 10x increase in data rates and size \(\Rightarrow\) HPC & CS
- Heterogeneous experiments & requirements \(\Rightarrow\) hotchpotch of math/CS solution
- Multi-modal data analysis, movies, ... \(\Rightarrow\) more complex reconstruction
- New experimental design \(\Rightarrow\) less regular data
Example: Learning Cell Identification from Spectral Data

Identify cell-type from concentration maps of P, Mn, Fe, Zn …
Learning Cell Identification via Nonnegative Matrix Factorization

\[
\minimize_{W,H} \| A - WH \|_F^2 \quad \text{subject to } W \geq 0, \ H \geq 0
\]

where “data” \( A \) is \( 1,000 \times 1,000 \) image \( \times 2,000 \) channels

- \( W \) are weight \( \approx \) additive elemental spectra
- \( H \) are images \( \approx \) additive elemental maps

Solve using (cheap) gradient steps ... need good initialization of \( W \)!

**Insight from Data**

Repeat analysis hundreds of times to, e.g., classify/identify cancerous cells etc.
Result: Learning Cell Identification from Spectral Data

Raw data...

... identify cell...

... classify cells

Traditional Cell Identification at APS
Ask student/postdoc to “mark” potential cell locations by hand & test

Opportunities for Applied Math & CS Light Sources
ML plus physical/statistical models, large-scale streaming data, ...
Outline

1. Data Analysis at DOE Light Sources

2. Optimization for Machine Learning
   - Mixed-Integer Nonlinear Optimization
     - Optimal Symbolic Regression
     - Deep Neural Nets as MIPs
     - Sparse Support-Vector Machines

3. Robust Optimization
   - Robust Optimization for SVMs

4. Conclusions and Extension
- Convexity & Sparsity-Inducing Norms
- Nonsmooth Optimization: Gradient, Subgradient & Proximal Methods
- Newton & Interior-Point Methods for ML
- Cutting-Pane Methods in ML
- Augmented Lagrangian Methods & ADMM
- Uncertainty & Robust optimization in ML
- (Inverse) Covariance Selection
Optimization for Machine Learning [Sra, Nowozin, & Wright (eds.)]
Optimization for Machine Learning [Sra, Nowozin, & Wright (eds.)]

- Convexity & Sparsity-Inducing Norms
- Nonsmooth Optimization: Gradient, Subgradient & Proximal Methods
- Newton & Interior-Point Methods for ML
- Cutting-Pane Methods in ML
- Augmented Lagrangian Methods & ADMM
- Uncertainty & Robust optimization in ML
- (Inverse) Covariance Selection

Important Argonne Legalese Disclaimer

I made zero contributions to this fantastic book! Worse: Until yesterday, I had no clue about this!!!
The Four Lands of Learning [Moritz Hardt, UC Berkeley]

Non-Convex Non-Optimization (2018 INFORMS Optimization Conference)
Non-Convex Non-Optimization (2018 INFORMS Optimization Conference)
The Four Lands of Learning [Moritz Hardt, UC Berkeley]

Non-Convex Non-Optimization (2018 INFORMS Optimization Conference)

Convexico

Gradientina

https://mrtz.org/gradientina.html#/
Non-Convex Non-Optimization (2018 INFORMS Optimization Conference)

Optopia

WE WANT OUR OPTOPIA NOW
Non-Convex Non-Optimization (2018 INFORMS Optimization Conference)

Optopia

https://mrtz.org/gradientina.html#/
Outline

1. Data Analysis at DOE Light Sources

2. Optimization for Machine Learning

3. Mixed-Integer Nonlinear Optimization
   - Optimal Symbolic Regression
   - Deep Neural Nets as MIPs
   - Sparse Support-Vector Machines

4. Robust Optimization
   - Robust Optimization for SVMs

5. Conclusions and Extension
Mixed-Integer Nonlinear Optimization

Mixed-Integer Nonlinear Program (MINLP)

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad c(x) \leq 0 \\
& \quad x \in \mathcal{X} \\
& \quad x_i \in \mathbb{Z} \text{ for all } i \in \mathcal{I}
\end{align*}
\]

... see survey, [Belotti et al., 2013]

- $\mathcal{X}$ bounded polyhedral set, e.g. $\mathcal{X} = \{ x : l \leq A^T x \leq u \}$
- $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ twice continuously differentiable (maybe convex)
- $\mathcal{I} \subset \{1, \ldots, n\}$ subset of integer variables
- MINLPs are NP-hard, see [Kannan and Monma, 1978]
- Worse: MINLP are undecidable, see [Jeroslow, 1973]
Goal in Optimal Symbolic Regression

Find symbolic mathematical expression that explains dependent variable in terms of independent variables *without assuming functional form!*

[Austel et al., 2017] propose MINLP model

- Find simplest symbolic mathematical expression ... *objective*
- Constrain the “grammar” of expressions ... *constraints*
- Match data (observations) to expression ... *continuous variables*
- Select “best” possible expression ... *binary variables*

... model mathematical expressions as a directed acyclic graph (DAG)
Factorable Functions and Expression Trees

**Definition (Factorable Function)**

\( f(x) \) is factorable iff expressed as sum of products of unary functions of a finite set
\( \mathcal{O}_{\text{unary}} = \{\sin, \cos, \exp, \log, | \cdot |\} \) whose arguments are variables, constants, or other functions, which are factorable.

- Combination of functions from set of operators
  \( \mathcal{O} = \{+, \times, /, ^{\wedge}, \sin, \cos, \exp, \log, | \cdot |\} \).
- Excludes integrals \( \int_{\xi=x_0}^{x} h(\xi) d\xi \) and black-box functions
- Can be represented as expression trees
- Forms basis for automatic differentiation & global optimization of nonconvex functions
  ... see, e.g. [Gebremedhin et al., 2005]

\[
f(x_1, x_2) = x_1 \log(x_2) + x_2^3
\]
Optimal Symbolic Regression [Austel et al., 2017]

- Build and solve optimal symbolic regression as MINLP
  - Form “supertree” of all possible expression trees
  - Use binary variables to switch parts of tree on/off
  - Compute data mismatch by propagating data values through tree
  - Minimize complexity (size) of expression tree with bound on data mismatch

⇒ large nonconvex MINLP model ... solved using Baron, SCIP, Couenne

Example: Kepler’s Law on planetary motion from NASA data with depth 3

<table>
<thead>
<tr>
<th>Data</th>
<th>2% Noise</th>
<th>10% Noise</th>
<th>30% Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex1</td>
<td>$\sqrt[3]{c\tau^2}M$</td>
<td>$\frac{\sqrt[3]{\tau^2}}{M + c}$</td>
<td>$\sqrt{c\tau^2}$</td>
</tr>
<tr>
<td>Ex2</td>
<td>$\sqrt[3]{c\tau^2}M$</td>
<td>$\sqrt[3]{\tau^2}c$</td>
<td>$\sqrt{\tau}$</td>
</tr>
<tr>
<td>Ex3</td>
<td>$\sqrt[3]{c\tau^2}M$</td>
<td>$\sqrt[3]{\tau M + \tau}$</td>
<td>$\sqrt{c\tau + c}$</td>
</tr>
</tbody>
</table>

Correct answer: $d = \sqrt[3]{\tau^2(M + m)}$ major semi-axis of $m$ orbiting $M$ at period $\tau$
Deep Neural Nets (DNNs) as MIPs [Fischetti and Jo, 2018]

Model DNN as MIP
- Model ReLU activation function with binary variables
- Model output of DNN as function of inputs (variable!)
- Solvable for DNNs of moderate size with MIP solvers

Image from Arden Dertad
Deep Neural Nets (DNNs) as MIPs [Fischetti and Jo, 2018]

Model DNN as MIP

- Model ReLU activation function with binary variables
- Model output of DNN as function of inputs (variable!)
- Solvable for DNNs of moderate size with MIP solvers

WARNING: Do not use for training of DNN!

MIP-model is totally unsuitable for training ... cumbersome & expensive to evaluate!
Deep Neural Nets (DNNs) as MIPs [Fischetti and Jo, 2018]

Model DNN as MIP

- Model ReLU activation function with binary variables
- Model output of DNN as function of inputs (variable!)
- Solvable for DNNs of moderate size with MIP solvers

WARNING: Do not use for training of DNN!

MIP-model is totally unsuitable for training ... cumbersome & expensive to evaluate!

Where can we use MIP models?

Use MIP for building adversarial examples that fool the DNN ... flexible!
Deep Neural Nets (DNNs) as MIPs [Fischetti and Jo, 2018]

- DNN with $K + 1$ layers: input = 0, ..., $K =$output
- $n_k$ nodes/units per layer $\text{UNIT}(j,k)$ with output $x_j^k \leftarrow \text{UNIT}(j,k)$
- $\text{UNIT}(j,k)$, e.g. ReLU: $x^k = \max(0, W^{k-1}x^{k-1} + b^{k-1})$, where $W^k, b^k$ DNN known parameters (from training)

**Key Insight (not new): Use Implication Constraints!**

Model $x = \max(0, w^T y + b)$ using implications, or binary variables:

$$x = \max(0, w^T y + b) \iff \begin{cases} w^T y + b = x - s, & x \geq 0, s \geq 0 \\ z \in \{0, 1\}, & \text{with } z = 1 \Rightarrow x \leq 0 \text{ and } z = 0 \Rightarrow s \leq 0 \end{cases}$$

... alternative $0 \leq s \perp x \geq 0$ complementarity constraint

Also model MaxPool: $x = \max(y_1, \ldots, y_t)$ using $t$ binary vars & SOS-1 constraint
Deep Neural Nets (DNNs) as MIPs [Fischetti and Jo, 2018]

Gives MIP model with flexible objective (DNN outputs $x^K$, binary vars $x$)

$$
\begin{array}{l}
\text{minimize} \quad c^T x + d^T z \\
\text{subject to} \quad \left( w_j^{k-1} \right)^T x^{k-1} + b_j^{k-1} = x_j^k - s_j^k, \quad x_j^k, s_j^k \geq 0 \\
\quad z_j^k \in \{0, 1\}, \quad \text{with} \quad z_j^k = 1 \Rightarrow x_j^k \leq 0 \quad \text{and} \quad z_j^k = 0 \Rightarrow s_j^k \leq 0 \\
\quad l^0 \leq x^0 \leq u^0
\end{array}
$$

... for given input = $x^0$, just compute output = $x^K$ expensive!

**Modeling Implication Constraints**

$$
\begin{align*}
&z \in \{0, 1\}, \quad \text{with} \quad z = 1 \Rightarrow x \leq 0 \quad \text{and} \quad z = 0 \Rightarrow s \leq 0 \\
&\iff z \in \{0, 1\}, \quad \text{with} \quad x \leq M_x (1 - z) \quad \text{and} \quad s \leq M_s z
\end{align*}
$$

**Use MIP for Building Adversarial Example**

- Fix weights $W, b$ from training data
- Find smallest perturbation to inputs $x^0$ that results in mis-classification
Deep Neural Nets (DNNs) as MIPs [Fischetti and Jo, 2018]

Example: DNN for digit classification as MIP

- Misclassify all digits: \( \hat{d} = (d + 5) \mod 10 \), i.e. \( 0 \to 5, 1 \to 6, \ldots \)
- Require activation of “wrong” digit in final layer is 20% above others
- Need tight bnds \( M_x, M_s \) in implications: propagate bnds forward through DNN

Results with CPLEX Solver and Tight Bounds (300s max CPU)

<table>
<thead>
<tr>
<th># Hidden</th>
<th># Nodes</th>
<th>% Solved</th>
<th># Nodes</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8</td>
<td>100</td>
<td>552</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>20/8</td>
<td>100</td>
<td>20,309</td>
<td>12.1</td>
</tr>
<tr>
<td>5</td>
<td>20/10</td>
<td>67</td>
<td>76,714</td>
<td>171.1</td>
</tr>
</tbody>
</table>

20 / 31
Sparse Support-Vector Machines

**Standard SVM Training**

- Data $S = \{x_i, y_i\}_{i=1}^m$: features $x_i \in \mathbb{R}^n$ labels $y_i \in \{-1, 1\}$
- $\xi \geq 0$ slacks, $b$ bias, $c > 0$ penalty parameter

$$\text{minimize}_{w, b, \xi} \frac{1}{2} \|w\|_2^2 + c\|\xi\|_1 = \frac{1}{2} \|w\|_2^2 + c1^T \xi$$

subject to $Y(Xw - b1) + \xi \geq 1$

$\xi \geq 0,$

where $Y = \text{diag}(y)$ and $X = [x_1, \ldots, x_m]^T$
Sparse Support-Vector Machines

**Standard SVM Training**

- **Data** $S = \{x_i, y_i\}_{i=1}^m$: features $x_i \in \mathbb{R}^n$ labels $y_i \in \{-1, 1\}$
- $\xi \geq 0$ slacks, $b$ bias, $c > 0$ penalty parameter

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|_2^2 + c\|\xi\|_1 = \frac{1}{2} \|w\|_2^2 + c1^T\xi \\
\text{subject to} & \quad Y(Xw - b) + \xi \geq 1 \\
& \quad \xi \geq 0,
\end{align*}
\]

where $Y = \text{diag}(y)$ and $X = [x_1, \ldots, x_m]^T$

**Find MINLP Model for Feature Selection in SVMs**

Given labeled training data find maximum margin classifier that minimizes hinge-loss and **cardinality of weight-vector**, $\|w\|_0$
Sparse Support-Vector Machines

[Guan et al., 2009] consider $\ell_0$-norm penalty on $w$ as MINLP

\[
\begin{aligned}
\text{minimize} & \quad \frac{1}{2} \|w\|_2^2 + a \|w\|_0 + c1^T \xi \\
\text{subject to} & \quad Y (Xw - b1) + \xi \geq 1, \ \xi \geq 0,
\end{aligned}
\]

Model $\ell_0$ with Perspective & Binary $z_j$ Counter

\[
\begin{aligned}
\text{minimize} & \quad 1^T u + a1^T z + c1^T \xi \\
\text{subject to} & \quad Y (Xw - b1) + \xi \geq 1, \ \xi \geq 0 \\
& \quad w_j^2 \leq z_j u_j, \ u \geq 0, \ z_j \in \{0, 1\}
\end{aligned}
\]

... conic-MIP, see, e.g. [Günlük and Linderoth, 2008]

... $w_j^2 \leq z_j u_j$ violates CQs $\Rightarrow$ weaker big-M formulation ...

\[
0 \leq u_j \leq M_u z_j, \quad w_j^2 \leq u_j
\]
Sparse Support-Vector Machines

[Goldberg et al., 2013] rewrite $w_j^2 \leq z_j u_j$ as

$$\| (2w_j, u_j - z_j) \|_2 \leq u_j + z_j$$

... second-order cone constraint ... and relax integrality ... add $\sum z_j \leq r$

... good classification accuracy & small $\| w \|_0$!
Sparse Support-Vector Machines [Maldonado et al., 2014]

Mixed-Integer Linear SVM

[Maldonado et al., 2014] formulate MILP: \( \min ||\xi||_1 \) subj. to \( ||w||_0 \leq B \)

\[
\begin{align*}
\text{minimize} & \quad 1^T \xi \\
\text{subject to} & \quad Y (Xw - b1) + \xi \geq 1 \\
& \quad Lz_j \leq w_j \leq Uz_j \\
& \quad \sum_j c_j z_j \leq B \\
& \quad \xi \geq 0, \quad z_j \in \{0, 1\}
\end{align*}
\]

for bounds \( L < U \) and budget \( B > 0 \)
Outline

1. Data Analysis at DOE Light Sources
2. Optimization for Machine Learning
3. Mixed-Integer Nonlinear Optimization
   - Optimal Symbolic Regression
   - Deep Neural Nets as MIPs
   - Sparse Support-Vector Machines
4. Robust Optimization
   - Robust Optimization for SVMs
5. Conclusions and Extension
Nonlinear Robust Optimization

**Nonlinear Robust Optimization**

minimize $f(x)$
subject to $c(x; u) \geq 0, \ \forall \ u \in \mathcal{U}$
$x \in \mathcal{X}$

**Small Example**

minimize $(x_1 - 4)^2 + (x_2 - 1)^2$
subject to $x_1 \sqrt{u} - x_2 u \leq 2,
\ldots \forall u \in [\frac{1}{4}, 2]$

Assumptions (e.g. [Leyffer et al., 2018]) ... wlog assume $f(x)$ is deterministic

- $u \in \mathcal{U}$ uncertain parameters closed convex set, independent of $x$
- $c(x; u) \geq 0 \ \forall \ u \in \mathcal{U}$ robust constraints ... semi-infinite optimization problem
- $\mathcal{X} \subset \mathbb{R}^n$ standard (certain) constraints; $f(x)$ and $c(x; u)$ smooth functions
Linear Robust Optimization [Ben-Tal and Nemirovski, 1999]

Robust linear constraints are easy! E.g. $a^T x + b \geq 0, \forall a \in \mathcal{U} := \{B^T a \geq c\}$

... rewrite semi-infinite constraint as a minimum

$$
\Leftrightarrow \left\{ \begin{array}{l}
\text{minimize} \ a^T x + b \\
\text{subject to} \ B^T a \geq c
\end{array} \right\} \geq 0
$$

... apply duality: $\mathcal{L}(a, \lambda) := a^T x + b - \lambda^T (B^T a - c)$

$$
\Leftrightarrow \left\{ \begin{array}{l}
\text{maximize} \ \mathcal{L}(a, \lambda) = b + \lambda^T c \\
\text{subject to} \ 0 = \nabla_a \mathcal{L}(a, \lambda) = x - B\lambda, \ \lambda \geq 0
\end{array} \right\} \geq 0
$$

... only need feasible point $\geq 0$ ... becomes standard polyhedral set

$$
b + \lambda^T c \geq 0, \ x = B\lambda, \ \lambda \geq 0$$
Duality Trick for Conic and Linear Robust Optimization

Duality trick generalizes to other conic uncertainty sets

\[
(P) \quad \text{minimize} \quad f(x) \quad \text{subject to} \quad c(x; u) \geq 0, \quad \forall \ u \in \mathcal{U}, \quad x \in \mathcal{X}
\]

... creates classes of tractable extended formulations

<table>
<thead>
<tr>
<th>Robust Constraints (c(x; u) \geq 0)</th>
<th>Uncertainty Set (\mathcal{U})</th>
<th>Extended Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>Polyhedral</td>
<td>Linear Program</td>
</tr>
<tr>
<td>Linear</td>
<td>Ellipsoidal</td>
<td>Conic QP</td>
</tr>
<tr>
<td>Conic</td>
<td>Conic</td>
<td>SDP</td>
</tr>
</tbody>
</table>
## Robust Optimization for Support Vector Machines (SVMs)

### Standard SVM Training
- Data $S = \{x_i, y_i\}_{i=1}^m$: features $x_i \in \mathbb{R}^n$ labels $y_i \in \{-1, 1\}$
- $\xi \geq 0$ slacks, $b$ bias, $c > 0$ penalty parameter

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}\|w\|_2^2 + c\mathbf{1}^T\xi \\
\text{subject to} & \quad Y (Xw - b\mathbf{1}) + \xi \geq 1, \quad \xi \geq 0,
\end{align*}
\]

where $Y = \text{diag}(y)$ and $X = [x_1, \ldots, x_m]^T$

### SVMs with Additive Location Errors
- See survey article [Caramanis et al., 2012] & use duality trick!
- Location errors $x_i^{\text{true}} = x_i + u_i$ & ellipsoid uncertainty $\mathcal{U} = \{u_i \mid u_i^T\Sigma u_i \leq 1\}$:

\[
\begin{align*}
y_i \left( w^T(x_i + u_i) - b \right) + \xi \geq 1, & \quad \forall u_i : u_i^T\Sigma u_i \leq 1 \\
y_i \left( w^Tx_i - b \right) + \xi + \|\Sigma^{1/2}w\|_2 \geq 1 & \quad \text{SOC constraint}
\end{align*}
\]
Robust Optimization for Support Vector Machines (SVMs)

**General Case of Location Errors: “Worst-Case SVM”**

\[
\begin{align*}
\text{minimize}_{w,b} \quad & \text{maximize}_{u \in U} \left\{ \frac{1}{2} \|w\|^2 + c \sum_j \max \left\{ 1 - y_j \left( w^T (x_j + u_j) - b \right), 0 \right\} \right\} \\
\text{for uncertainty set } U = \left\{ (u_1, \ldots, u_m) \mid \sum_j \|u_j\| \leq d \right\} \text{ equivalent to} \\
\text{minimize}_{w,b} \left\{ \frac{1}{2} \|w\|^2 + d \|w\|_D + c \sum_j \max \left\{ 1 - y_j \left( w^T (x_j + u_j) - b \right), 0 \right\} \right\}
\end{align*}
\]

where \( \| \cdot \|_D \) is dual norm of \( \| \cdot \| \), e.g. \( \ell_2 \leftrightarrow \ell_2 \) or \( \ell_\infty \leftrightarrow \ell_1 \), ... follows from duality

[Caramanis et al., 2012] argue that derivation shows that:

- Regularized classifiers are more robust: satisfy worst-case principle
- Provide probabilistic interpretation if viewed as chance constraints
Conclusions and Extension: Optimization for Machine Learning

Conclusions

- **Mixed-Integer Optimization for Machine Learning**
  - Optimal symbolic regression, expression trees, nonconvex MIP
  - MIPs of deep neural nets for building adversarial examples
  - Support-vector machines & $\ell_0$ regularizers & constraints
- **Robust Optimization for Machine Learning**
  - Best “worst-case” SVM $\Rightarrow$ equivalent tractable formulation

Extensions and Challenges

- Extending use of integer variables into design of DNNs
- Realistic stochastic interpretation of regularizers in SVM, DNN, ...


*Optimization for machine learning*, page 369.

*Constraints*, pages 1–14.


In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 450–457. SIAM.


