Optimization for Machine Learning

LANS Informal Seminar

Sven Leyffer

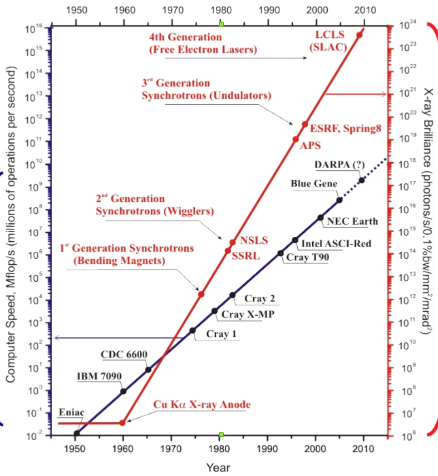
Argonne National Laboratory

November, 28 2018

Outline

Motivation: Datanami from DOE Lightsource Upgrades

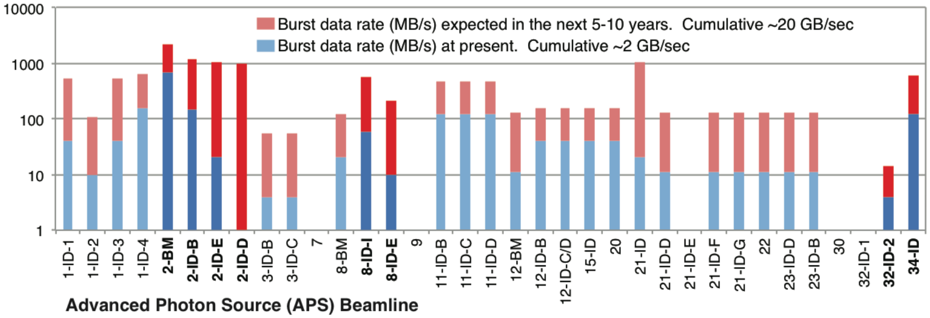
Computers:
12 orders of magnitude in 6 decades



Light sources:
18 orders of magnitude in 5 decades

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

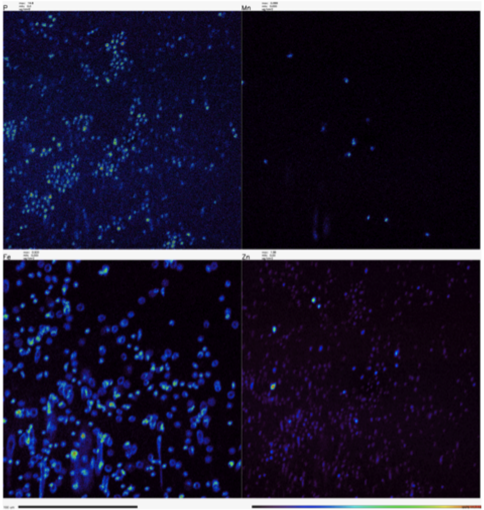
Challenges at DOE Lightsources



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



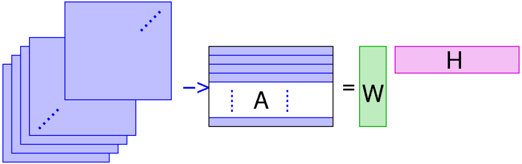
Identify cell-type from concentration maps of P, Mn, Fe, Zn ...

Learning Cell Identification via Nonnegative Matrix Factorization

$$\underset{W,H}{\text{minimize}} \ \|A - WH\|_F^2 \quad \text{subject to } W \geq 0, \ H \geq 0$$

where "data" A is $1,000 \times 1,000$ image $\times 2,000$ channels

- W are weight $\simeq$ additive elemental spectra
- H are images $\simeq$ additive elemental maps

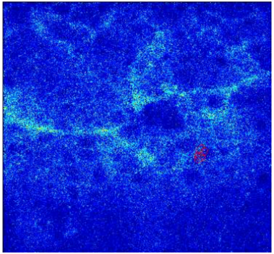


Solve using (cheap) gradient steps ... need good initialization of W!

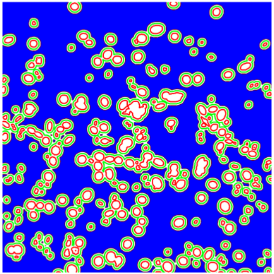
Insight from Data

Repeat analysis hundreds of times to, e.g., classify/identify cancerous cells etc.

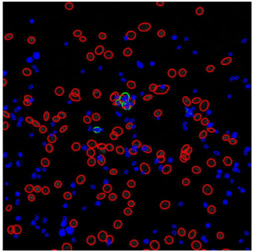
Result: Learning Cell Identification from Spectral Data



Raw data ...



... identify cell ...



... classify cells

Traditional Cell Identification at APS

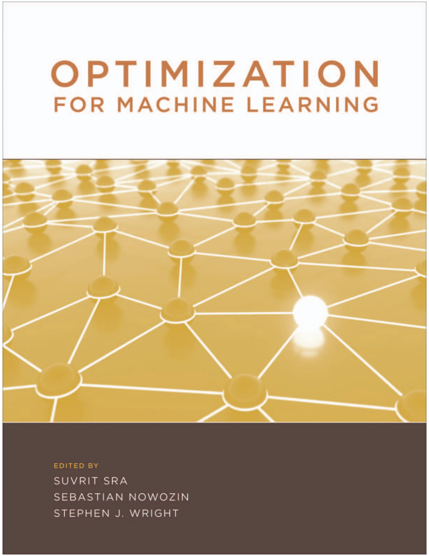
Ask student/postdoc to “mark” potential cell locations by hand & test

Opportunities for Applied Math & CS Light Sources

ML plus physical/statistical models, large-scale streaming data, ...

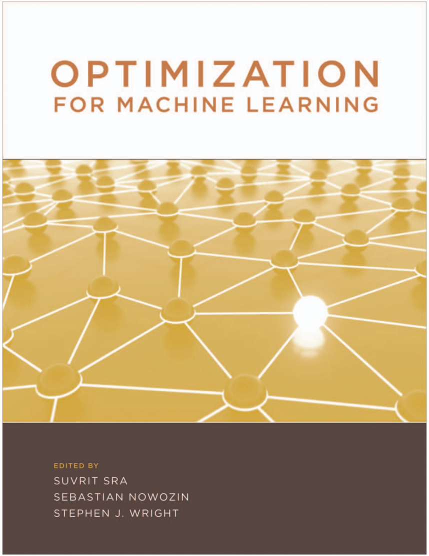
Outline

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

Optimization for Machine Learning [Sra, Nowozin, & Wright (eds.)]

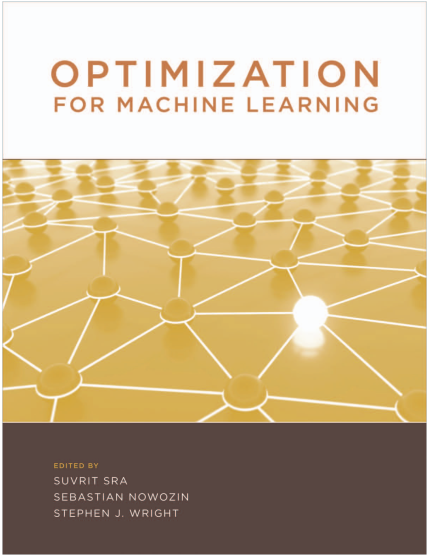


- Convexity & Sparsity-Inducing Norms
- Nonsmooth Optimization: Gradient, Subgradient & Proximal Methods
- Newton & Interior-Point Methods for ML
- Cutting-Pane Methods in ML
- Augmented Lagrangian Methods & ADMM
- Uncertainty & Robust optimization in ML
- (Inverse) Covariance Selection

Important Argonne Legalese Disclaimer

I made zero contributions to this fantastic book!

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 this fall, I had no clue about this!!!

The Four Lands of Learning [Moritz Hardt, UC Berkeley]

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

The Four Lands of Learning [Moritz Hardt, UC Berkeley]

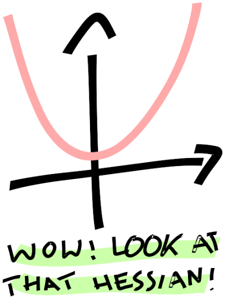
Non-Convex Non-Optimization (2018 INFORMS Optimization Conference)

Convexico

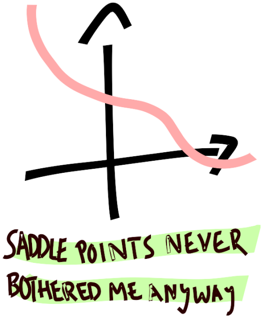
The Four Lands of Learning [Moritz Hardt, UC Berkeley]

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

Convexico



Gradientina



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

The Four Lands of Learning [Moritz Hardt, UC Berkeley]

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

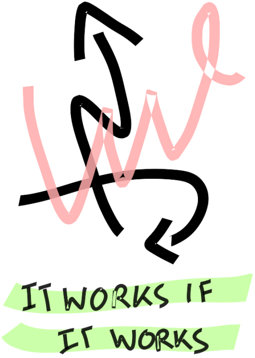
Optopia

The Four Lands of Learning [Moritz Hardt, UC Berkeley]

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

Optopia

Messiland



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

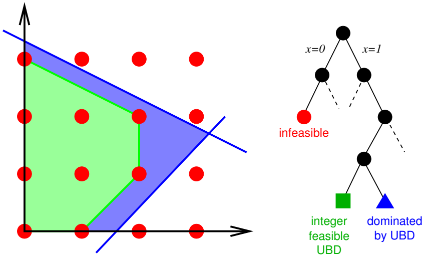
Outline

Mixed-Integer Nonlinear Optimization

Mixed-Integer Nonlinear Program (MINLP)

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

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



- $\mathcal{X}$ bounded polyhedral set, e.g. $\mathcal{X} = \{x \ : \ l \leq A^T x \leq u\}$
- $f : \mathbb{R}^n \to R$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ 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]

Optimal Symbolic Regression

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)

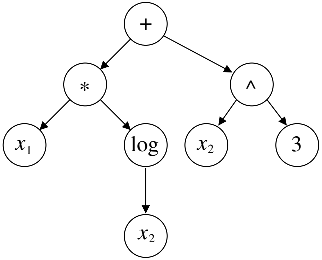
Factorable Functions and Expression Trees

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 $\mathcal{O} = \{+, \times, /, \hat{}, \sin, \cos, \exp, \log, |\cdot|\}$.
- 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]

Expression Tree



$f(x_1, x_2) = x_1 \log(x_2) + x_2^3$

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

$\Rightarrow$ large nonconvex MINLP model ... solved using `Baron`, `SCIP`, `Couenne`

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

Data	2% Noise	10% Noise	30% Noise
Ex1	$\sqrt[3]{c\tau^2 M}$	$\sqrt[3]{\tau^2}(M+c)$	$\sqrt{c\tau^2}$
Ex2	$\sqrt[3]{c\tau^2 M}$	$\sqrt[3]{\tau^2}c$	$\sqrt{\tau}$
Ex3	$\sqrt[3]{c\tau^2 M}$	$\sqrt[3]{\tau M}+\tau$	$\sqrt{c\tau}+c$

Correct answer: $d = \sqrt[3]{c\tau^2(M+m)}$ major semi-axis of m orbiting M at period τ

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

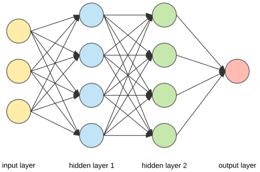


Image from Arden Dertad

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

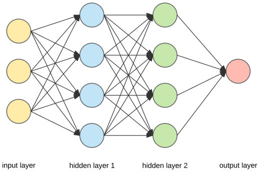


Image from Arden Dertad

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

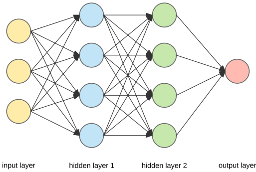


Image from Arden Dertad

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$=0,\ldots,K=$output
- n_k nodes/units per layer `UNIT(j,k)` with output $x_j^k \leftarrow$ `UNIT(j,k)`
- `UNIT(j,k)`, e.g. ReLU: $x^k = \max\left(0, W^{k-1}x^{k-1} + b^{k-1}\right)$, 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:

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

... 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)

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

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

Modeling Implication Constraints

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

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

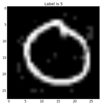
Deep Neural Nets (DNNs) as MIPs [Fischetti and Jo, 2018]

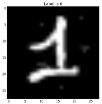
Example: DNN for digit classification as MIP

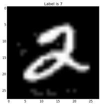
- Misclassify all digits: $\hat{d} = (d + 5) \bmod 10$, i.e. $0 \to 5$, $1 \to 6$, ...
- Require activation of "wrong" digit in final layer is 20% above others
- Need tight bnds M_x, M_s in implications: propagate bnds forward through DNN

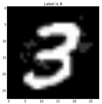
Results with CPLEX Solver and Tight Bounds (300s max CPU)

# Hidden	# Nodes	% Solved	# Nodes	CPU
3	8	100	552	0.6
4	20/8	100	20,309	12.1
5	20/10	67	76,714	171.1









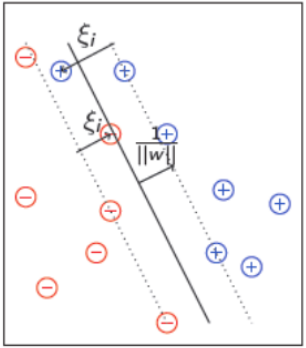
Sparse Support-Vector Machines

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

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}} \quad & \tfrac{1}{2}\|w\|_2^2 + c\|\xi\|_1 = \tfrac{1}{2}\|w\|_2^2 + c\mathbf{1}^T\xi \\ \text{subject to} \quad & Y(Xw - b\mathbf{1}) + \xi \geq \mathbf{1} \\ & \xi \geq 0, \end{aligned}$$

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



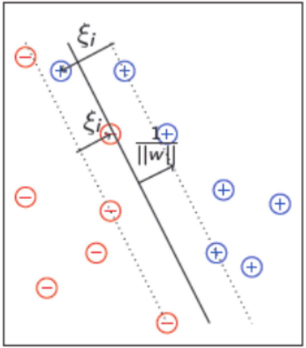
Sparse Support-Vector Machines

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

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}} \quad & \tfrac{1}{2}\|w\|_2^2 + c\|\xi\|_1 = \tfrac{1}{2}\|w\|_2^2 + c\mathbf{1}^T\xi \\ \text{subject to } & Y(Xw - b\mathbf{1}) + \xi \geq \mathbf{1} \\ & \xi \geq 0, \end{aligned}$$

where $Y = \mathrm{diag}(y)$ 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

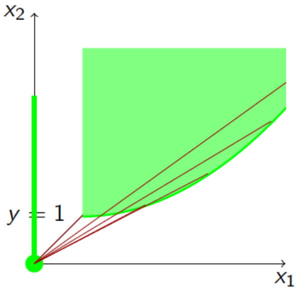
[Guan et al., 2009] consider ℓ_0-norm penalty on w as MINLP

$$\begin{aligned} \underset{w,b,\xi}{\text{minimize}} \quad & \tfrac{1}{2}\|w\|_2^2 + a\|w\|_0 + c\mathbf{1}^T\xi \\ \text{subject to } & Y(Xw - b\mathbf{1}) + \xi \geq \mathbf{1},\ \xi \geq 0, \end{aligned}$$

Model ℓ_0 with Perspective & Binary z_j Counter

$$\begin{aligned} \underset{u,w,b,\xi,z}{\text{minimize}} \quad & \mathbf{1}^T u + a\mathbf{1}^T z + c\mathbf{1}^T\xi \\ \text{subject to } & Y(Xw - b\mathbf{1}) + \xi \geq \mathbf{1},\ \xi \geq 0 \\ & w_j^2 \leq z_j u_j,\ u \geq 0,\ z_j \in \{0,1\} \end{aligned}$$

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



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

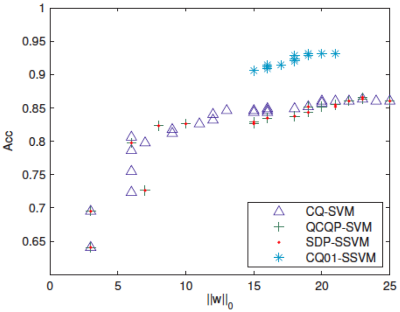
$$0 \leq u_j \leq M_u z_j, \quad w_j^2 \leq u_j$$

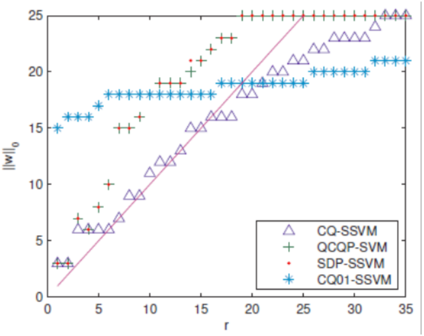
Sparse Support-Vector Machines

[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 $\|w\|_0 \leq B$

$$\begin{aligned} \underset{w,b,\xi,z}{\text{minimize}} \quad & \mathbf{1}^T \xi && \text{classification error} \\ \text{subject to } & Y(Xw - b\mathbf{1}) + \xi \geq \mathbf{1} && \text{classifier c/s} \\ & L z_j \leq w_j \leq U z_j && \text{on/off } w_j \\ & \sum_j c_j z_j \leq B && \text{budget constraint} \\ & \xi \geq 0, \quad z_j \in \{0, 1\} \end{aligned}$$

for bounds $L < U$ and budget $B > 0$

Outline

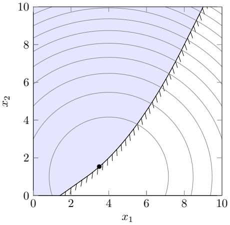
Nonlinear Robust Optimization

Nonlinear Robust Optimization

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to } & c(x; u) \geq 0, \quad \forall\, u \in \mathcal{U} \\ & x \in \mathcal{X} \end{aligned}$$

Small Example

$$\begin{aligned} \underset{x \geq 0}{\text{minimize}} \quad & (x_1 - 4)^2 + (x_2 - 1)^2 \\ \text{subject to } & x_1\sqrt{u} - x_2 u \leq 2, \\ & \ldots \forall u \in \left[\tfrac{1}{4}, 2\right] \end{aligned}$$



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. $a^T x + b \geq 0, \quad \forall a \in \mathcal{U} := \{B^T a \geq c\}$

... rewrite semi-infinite constraint as a minimum

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

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

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

... only need feasible point ≥ 0 ... becomes standard polyhedral set

$$b + \lambda^T c \geq 0, \quad x = B\lambda, \quad \lambda \geq 0$$

Duality Trick for Conic and Linear Robust Optimization

Duality trick generalizes to other conic uncertainty sets

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

... creates classes of tractable extended formulations

Robust Constraints $c(x; u) \geq 0$	Uncertainty Set $\mathcal{U}$	Extended Formulation
Linear	Polyhedral	Linear Program
Linear	Ellipsoidal	Conic QP
Conic	Conic	SDP

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

$$\begin{aligned} &\underset{w,b,\xi}{\text{minimize}} \quad \tfrac{1}{2}\|w\|_2^2 + c\mathbf{1}^T\xi \\ &\text{subject to } Y\left(Xw - b\mathbf{1}\right) + \xi \geq \mathbf{1}, \qquad \xi \geq 0, \end{aligned}$$

where $Y = \operatorname{diag}(y)$ 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 $\mathcal{U} = \{u_i \mid u_i^T \Sigma u_i \leq 1\}$:

$$\begin{aligned} & y_i\left(w^T(x_i + u_i) - b\right) + \xi \geq 1, \qquad \forall u_i : u_i^T \Sigma u_i \leq 1 \\ \Leftrightarrow\ & y_i\left(w^T x_i - b\right) + \xi - \|\Sigma^{1/2} w\|_2 \geq 1 \qquad \text{SOC constraint} \end{aligned}$$

Robust Optimization for Support Vector Machines (SVMs)

General Case of Location Errors: "Worst-Case SVM"

$$\underset{w,b}{\text{minimize}} \ \underset{u \in \mathcal{U}}{\text{maximize}} \ \left\{ \frac{1}{2}\|w\|_2^2 + c \sum_j \max\left\{1 - y_j\left(w^T(x_j + u_j) - b\right), 0\right\} \right\}$$

for uncertainty set $U = \left\{(u_1, \ldots, u_m) \mid \sum_j \|u_j\| \leq d\right\}$ equivalent to

$$\underset{w,b}{\text{minimize}} \ \left\{ \frac{1}{2}\|w\|_2^2 + d\|w\|_D + c \sum_j \max\left\{1 - y_j\left(w^T(x_j + u_j) - b\right), 0\right\} \right\}$$

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

Illustration of Robust Machine-Learning

Standard ML Uses Subset of Data

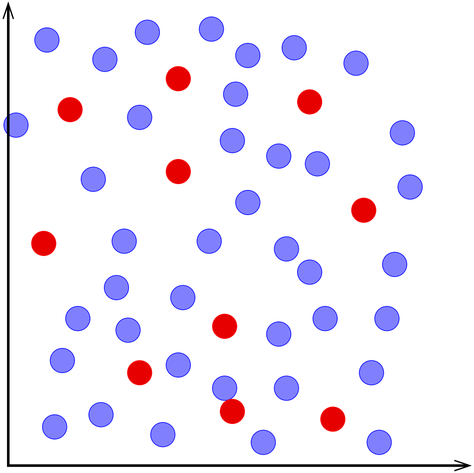
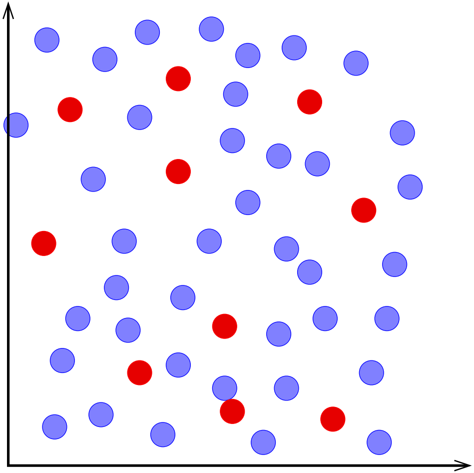
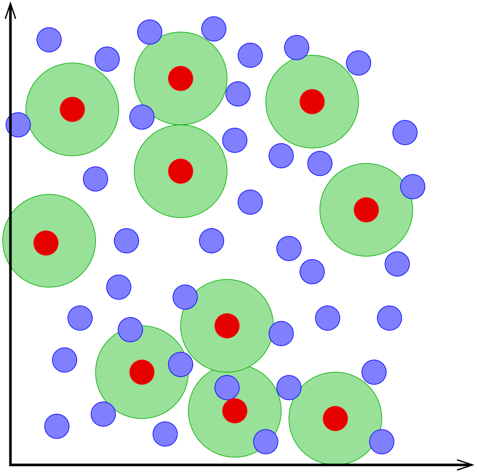


Illustration of Robust Machine-Learning

Standard ML Uses Subset of Data

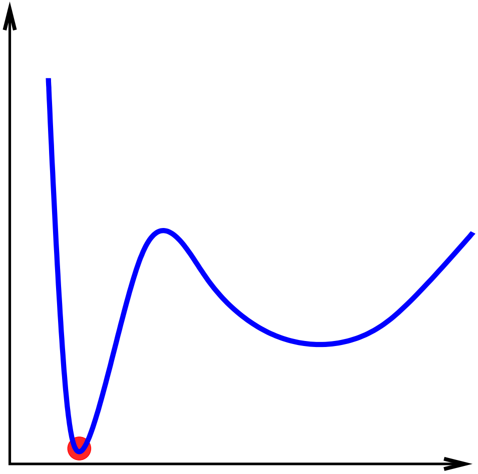


Robust ML Uses Region of Data



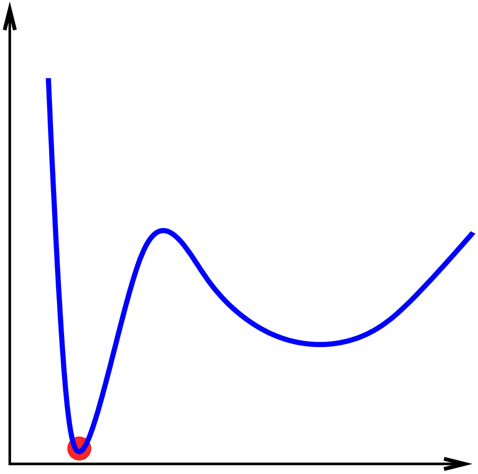
Global vs. Robust Minimizers

Global Minimizer not Robust wrt Perturbations

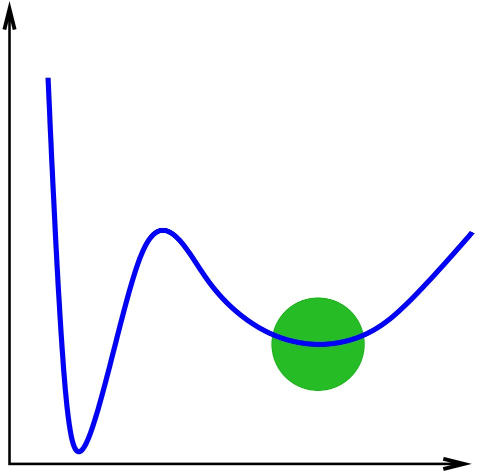


Global vs. Robust Minimizers

Global Minimizer not Robust wrt Perturbations



Robust Minimizer is Robust wrt Perturbations



Outline

Stochastic Gradient Descend et al. [Bottou et al, SIREV 60(2), 2018]

Great intro to Optimization for ML

1. Analysis of Stochastic Gradient
2. Noise Reduction
3. Newton & 2^{nd} Order Methods

A Must-Read Paper!

SIAM Review
Vol. 60, No. 2, pp. 223–311

Optimization Methods for Large-Scale Machine Learning*

Léon Bottou†
Frank E. Curtis‡
Jorge Nocedal§

Abstract. This paper provides a review and commentary on the past, present, and future of numerical optimization algorithms in the context of machine learning applications. Through case studies on text classification and the training of deep neural networks, we discuss how optimization problems arise in machine learning and what makes them challenging. A major theme of our study is that large-scale machine learning represents a distinctive setting in which the stochastic gradient (SG) method has traditionally played a central role while conventional gradient-based nonlinear optimization techniques typically falter. Based on this viewpoint, we present a comprehensive theory of a straightforward, yet versatile SG algorithm, discuss its practical behavior, and highlight opportunities for designing algorithms with improved performance. This leads to a discussion about the next generation of optimization methods for large-scale machine learning, including an investigation of two main streams of research on techniques that diminish noise in the stochastic directions and methods that make use of second-order derivative approximations.

Key words. numerical optimization, machine learning, stochastic gradient methods, algorithm complexity analysis, noise reduction methods, second-order methods

AMS subject classifications. 65K05, 68Q25, 68T05, 90C06, 90C30, 90C90

DOI. 10.1137/16M1080173

Stochastic Gradient Descend et al. [Bottou et al, SIREV 60(2), 2018]

Generic Training/Optimization Problem in ML

$$\underset{w}{\text{minimize}} \;\; F(w) = \mathbb{E}_\xi\left[f(w;\xi)\right] \quad \text{or} \quad = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

Stochastic Gradient Method (starting at w_1)

for k=1,2,... **do**

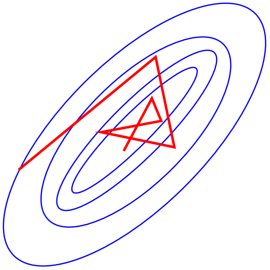
  Generate a realization of random variable ξ_k (select i_k)

  Get stochastic gradient vector $g(w_k,\xi_k)$, e.g. $\nabla_w f(w_k;\xi_k)$

  Choose stepsize $\alpha_k > 0$

  New iterate $w_{k+1} \leftarrow w_k - \alpha_g g(w_k,\xi_k)$

end



Stochastic Gradient Descend et al. [Bottou et al, SIREV 60(2), 2018]

Sloppy Theorem: Convergence of Stochastic Gradient for Strictly Convex $F(w)$

Assume fixed stepsize $0 < \hat{\alpha} \leq \frac{\mu}{LM}$ then for all $k = 1, 2, \ldots$

$$\mathbb{E}\left[F(w_k) - F(w^*)\right] \to \frac{\hat{\alpha} LM}{2c\mu} \qquad \text{expected optimality gap}$$

where

- μ satisfies $\nabla F(w_k)^T \mathbb{E}\left[g(w_k;\xi_k)\right] \geq \mu \|\nabla F(w_k)\|_2^2$
- L Lipschitz constant: $\|\nabla F(w) - \nabla F(\hat{w})\|_2 \leq L\|w - \hat{w}\|_2, \quad$ for all $w, \hat{w}$
- M second-moment bound: $\mathbb{E}\left[\|g(w_k;\xi_k)\|_2^2\right] \leq M + M\|\nabla F(w_k)\|_2^2$
- c strong convexity const.: $F(\hat{w}) \geq F(w) + \nabla F(w)^T (w - \hat{w}) + c\|\hat{w} - w\|_2^2$

... convergence to neighborhood of solution, only!

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 & ℓ_0 regularizers & constraints
- Robust Optimization for Machine Learning
  - Best "worst-case" SVM $\Rightarrow$ equivalent tractable formulation
- Stochastic Gradient Descend and Convergence in Expectation

Extensions and Challenges

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

Austel, V., Dash, S., Gunluk, O., Horesh, L., Liberti, L., Nannicini, G., and Schieber, B. (2017).
Globally optimal symbolic regression.
arXiv preprint arXiv:1710.10720.

Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., and Mahajan, A. (2013).
Mixed-integer nonlinear optimization.
Acta Numerica, 22:1–131.

Ben-Tal, A. and Nemirovski, A. (1999).
Robust solutions of uncertain linear programs.
Operations Research Letters, 25(1):1–13.

Caramanis, C., Mannor, S., and Xu, H. (2012).
14 robust optimization in machine learning.
Optimization for machine learning, page 369.

Fischetti, M. and Jo, J. (2018).
Deep neural networks and mixed integer linear optimization.
Constraints, pages 1–14.

Gebremedhin, A. H., Manne, F., and Pothen, A. (2005).
What color is your jacobian? graph coloring for computing derivatives.
SIAM review, 47(4):629–705.

Goldberg, N., Leyffer, S., and Munson, T. (2013).
A new perspective on convex relaxations of sparse svm.
In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 450–457. SIAM.

Guan, W., Gray, A., and Leyffer, S. (2009).
Mixed-integer support vector machine.
In *NIPS workshop on optimization for machine learning*.

Günlük, O. and Linderoth, J. (2008).
Perspective relaxation of mixed integer nonlinear programs with indicator variables.
In Lodi, A., Panconesi, A., and Rinaldi, G., editors, *IPCO 2008: The Thirteenth Conference on Integer Programming and Combinatorial Optimization*, volume 5035, pages 1–16.

Jeroslow, R. G. (1973).
There cannot be any algorithm for integer programming with quadratic constraints.
Operations Research, 21(1):221–224.

Kannan, R. and Monma, C. (1978).
On the computational complexity of integer programming problems.
In Henn, R., Korte, B., and Oettli, W., editors, *Optimization and Operations Research,*, volume 157 of *Lecture Notes in Economics and Mathematical Systems*, pages 161–172. Springer.

Leyffer, S., Menickelly, M., Munson, T., Vanaret, C., , and Wild, S. M. (2018).
Nonlinear robust optimization.
Technical report, Argonne National Laboratory, Mathematics and Computer Science Division.

Maldonado, S., Pérez, J., Weber, R., and Labbé, M. (2014).
Feature selection for support vector machines via mixed integer linear programming.
Information sciences, 279:163–175.