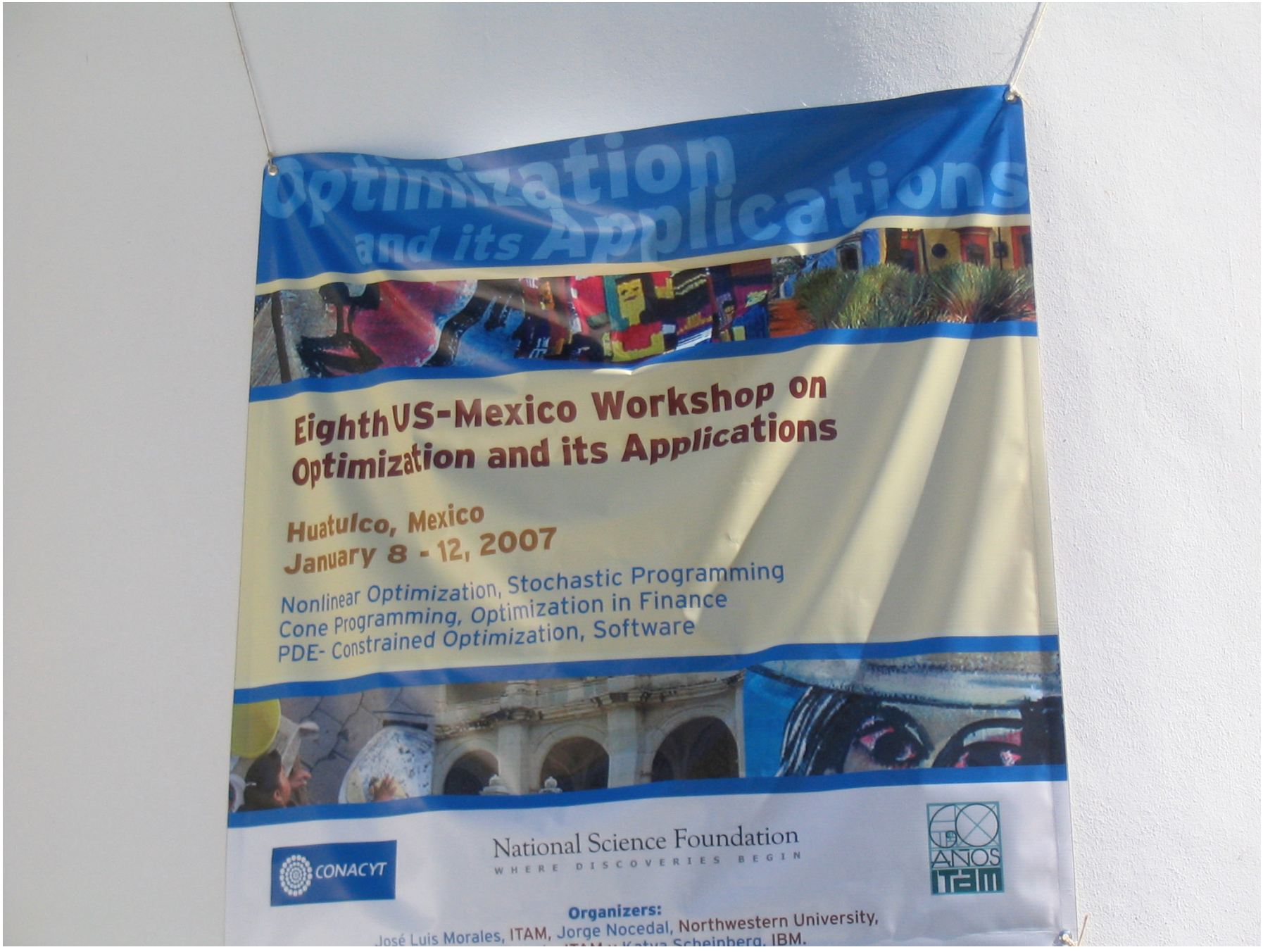
That's just, like, uh, your opinion, man

Stephen Wright

University of Wisconsin-Madison

Huatulco, January 2023.


Optimization and its Applications
Eighth US-Mexico Workshop on Optimization and its Applications
Huatulco, Mexico
January 8 - 12, 2007
Nonlinear Optimization, Stochastic Programming
Cone Programming, Optimization in Finance
PDE- Constrained Optimization, Software
CONACYT
National Science Foundation
WHERE DISCOVERIES BEGIN
ANOS
ITAM
Organizers:
José Luis Morales, ITAM, Jorge Nocedal, Northwestern University,

Applications with ℓ_1-Norm Objective Terms

Stephen Wright

University of Wisconsin-Madison

Huatulco, January 2007

1. Formulation

2. Least-Squares with ℓ_1
   - Applications
   - Algorithms
   - Results

3. Logistic Regression
   - Application
   - Algorithms
   - Results

Based on joint work with Weiliang Shi, Grace Wahba, Rob Nowak, Mario Figueiredo.

Formulation

We describe two classes of applications for problems of the form:

$$\min_x \ f(x) + \lambda \|x\|_1$$

where $x \in \mathbb{R}^n$; f is convex, smooth, possibly nonlinear; $\lambda > 0$ is a regularization parameter.

A special case of particular interest:

$$\min_x \ \frac{1}{2}\|Ax - y\|_2^2 + \lambda \|x\|_1$$

- n may be very large (hence, storage and computational limitations);
- ℓ_1 norm may apply to only a subvector of x;
- may wish to solve for a number of λ values.

Use well-known optimization techniques, tailored to structure and characteristics of the applications.

	1/8/2007 MONDAY		1/9/2007 TUESDAY		1/10/2007 WED	1/11/2007 THURSDAY		1/12/2007 FRIDAY	
8:45-9:00	*Chair*	**Opening**							
9:00 - 9:40	*Wright*	**M. Powell**	*Toint*	**A. Waechter**	*free day*	*Leyffer*	**Wright**	9:00 - 9:40	*Byrd* **M. Todd**
9:40 - 10:05		**Leyffer**		**Flores**	*excursion*		**Walther**	9:40 - 10:05	**Steihaug**
10:05 - 10:30		**Parada**		**Gay**			**Biros**	10:05 - 10:20	break
10:30 - 10:50		break		break			break	10:20 - 11:00	**Pena**
10:50 - 11:15		**Zhang**		**Nocedal**			**Tapia**	11:00- 11:25	**Waltz**
11:15 - 11:40		**Todorov**		**Curtis**			**Dominguez**	workshop ends at 11:30	
11:40 - 12:05		**Ferris**		**Zhu**			**Scheinberg**		
12:05 - 16:00		mid-day break		mid-day break			mid-day break		
16:00 - 16:40	*Overton*	**Morales**	*Biegler*	**Overton**		*Ferris*	**Goldfarb**		
16:40 - 17:05		**Byrd**		**Boggs**			**Gunluk**		
17:05 - 17:30		**Toint**		**Plantenga**			**Burke**		
17:30 - 17:50		break		break			break		
17:50 - 18:15		**Biegler**		**Guerra**			**Barrera**		
18:15 - 18:40		**Villalobos**		**Hintermeuller**			**Conn**		
18:40 - 19:05		**Benson**		**Haber**			**Lopez-Calva**		

Origins

My talk here in Jan 2007 was my first on sparse optimization / compressed sensing. (Yin Zhang spoke on this topic at the same meeting.)

It quickly became a major topic of interest in optimization, with many participants - some already working in optimization and some from outside.

(Sparse optimization was listed as a topic of interest at the 2009 ISMP, 2011 SIAM Conference on Optimization, etc.)

Led naturally to a wider engagement with ML.

Also, built on many earlier works by optimization people on ML, e.g.

- Least squares (linear and nonlinear; Tikhonov regularization). Robust regression (e.g. Huber estimator).
- Olvi Mangasarian [Mangasarian, 1965, Mangasarian, 1968]
- Kernel SVM [Fine and Scheinberg, 2001], [Ferris and Munson, 2002], [Gertz and Wright, 2003]
- SW with Grace Wahba's group [Lu et al., 2005, Shi et al., 2008]

Theme

What influence has machine learning had on optimization?

- research directions and themes;
- practices and culture.

The encounter with ML has led to increased interest in optimization, among a wide community with high visibility (in both the scientific community and the general public).

It has significantly increased the range and number of people writing papers on optimization.

Caveat. A personal perspective, based on incomplete data. Your experience may be deeper and more varied!

I. Paradigms / Formulations

ML has highlighted several paradigms that were not previously mainstream.

- support vector machines (linear and kernel SVM);
- finite-sum structure;
- regularization: ℓ_1, TV, group-sparse, nonconvex;
- hyperparameter optimization;
- matrix optimization (involving low-rank and sparse matrices);
- neural network training.

Linear Least Squares

$$\min_x f(x) := \frac{1}{2}\sum_{j=1}^{m}(a_j^T x - y_j)^2 = \frac{1}{2}\|Ax - y\|_2^2.$$

[Gauss, 1799], [Legendre, 1805]. We thought this was a solved problem in numerical linear algebra (Saunders, ...) but it's still a popular topic in ML!

- ℓ_2 regularization reduces sensitivity of the solution x to noise in y.

$$\min_x \frac{1}{2}\|Ax - y\|_2^2 + \lambda\|x\|_2^2.$$

- ℓ_1 regularization (LASSO) [Tibshirani, 1996] yields sparse solutions.

$$\min_x \frac{1}{2}\|Ax - y\|_2^2 + \lambda\|x\|_1.$$

  Feature selection: Nonzero locations in x indicate important components of feature vectors a_j.

Initially, ℓ_1 was solved as a QP parametrized by λ e.g. [Efron et al., 2004].

Application to compressed sensing resulted in many new methods.

Linear SVM

Each item of data belongs to one of two classes: $y_j = +1$ and $y_j = -1$.

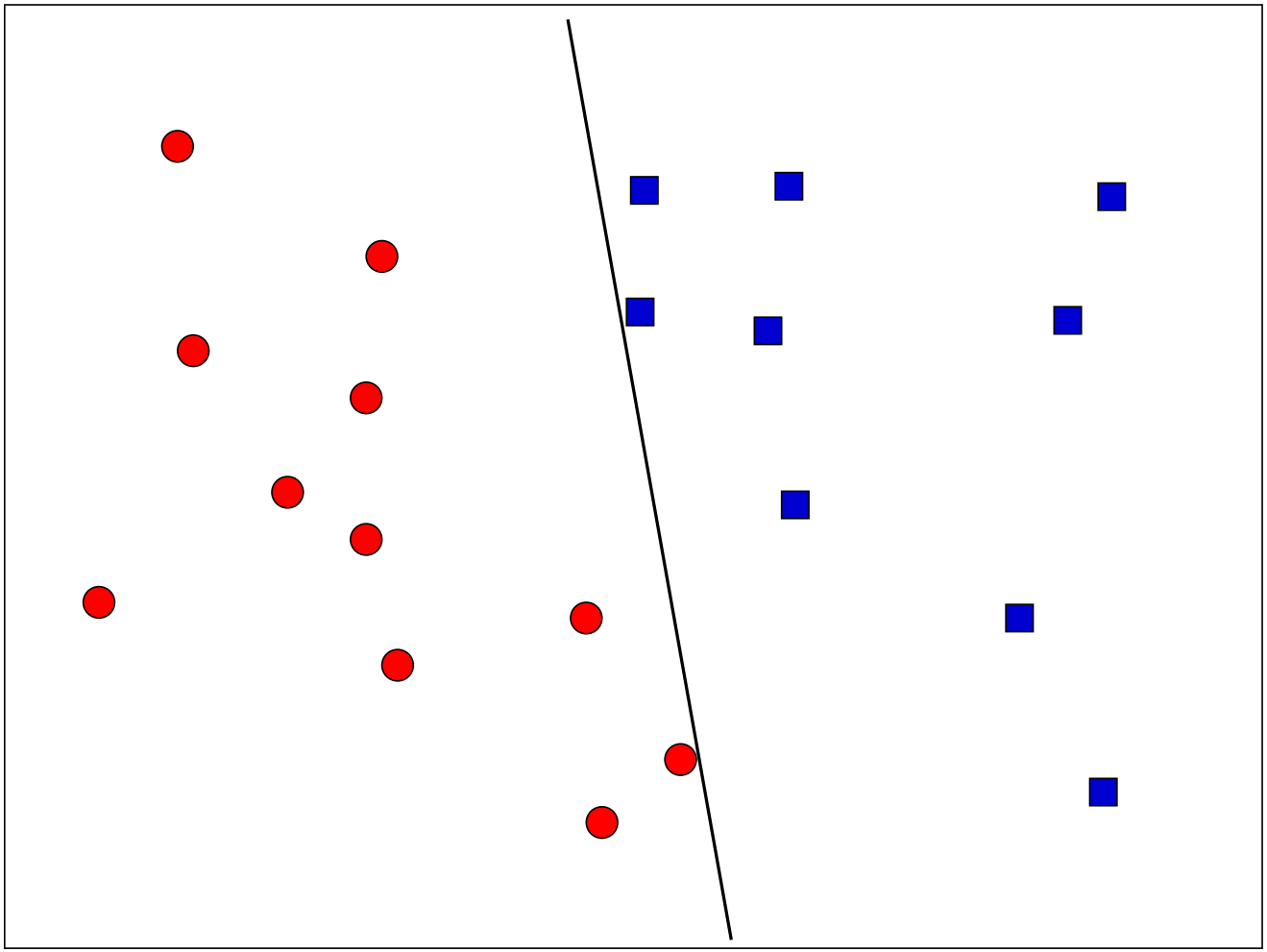
Seek (x, β) such that

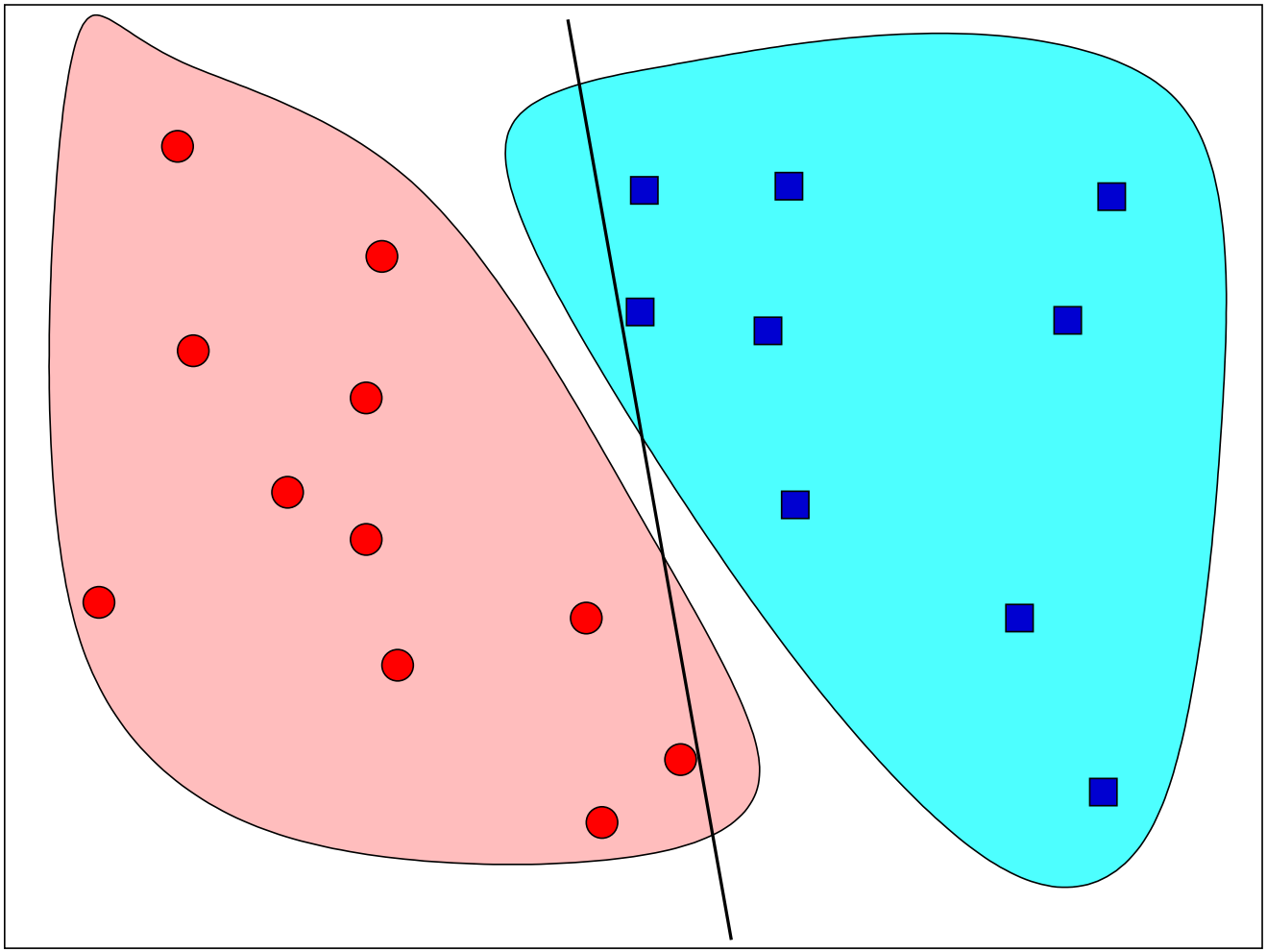
$$\begin{aligned} a_j^T x - \beta &\geq 1 \quad \text{when } y_j = +1; \\ a_j^T x - \beta &\leq -1 \quad \text{when } y_j = -1. \end{aligned}$$

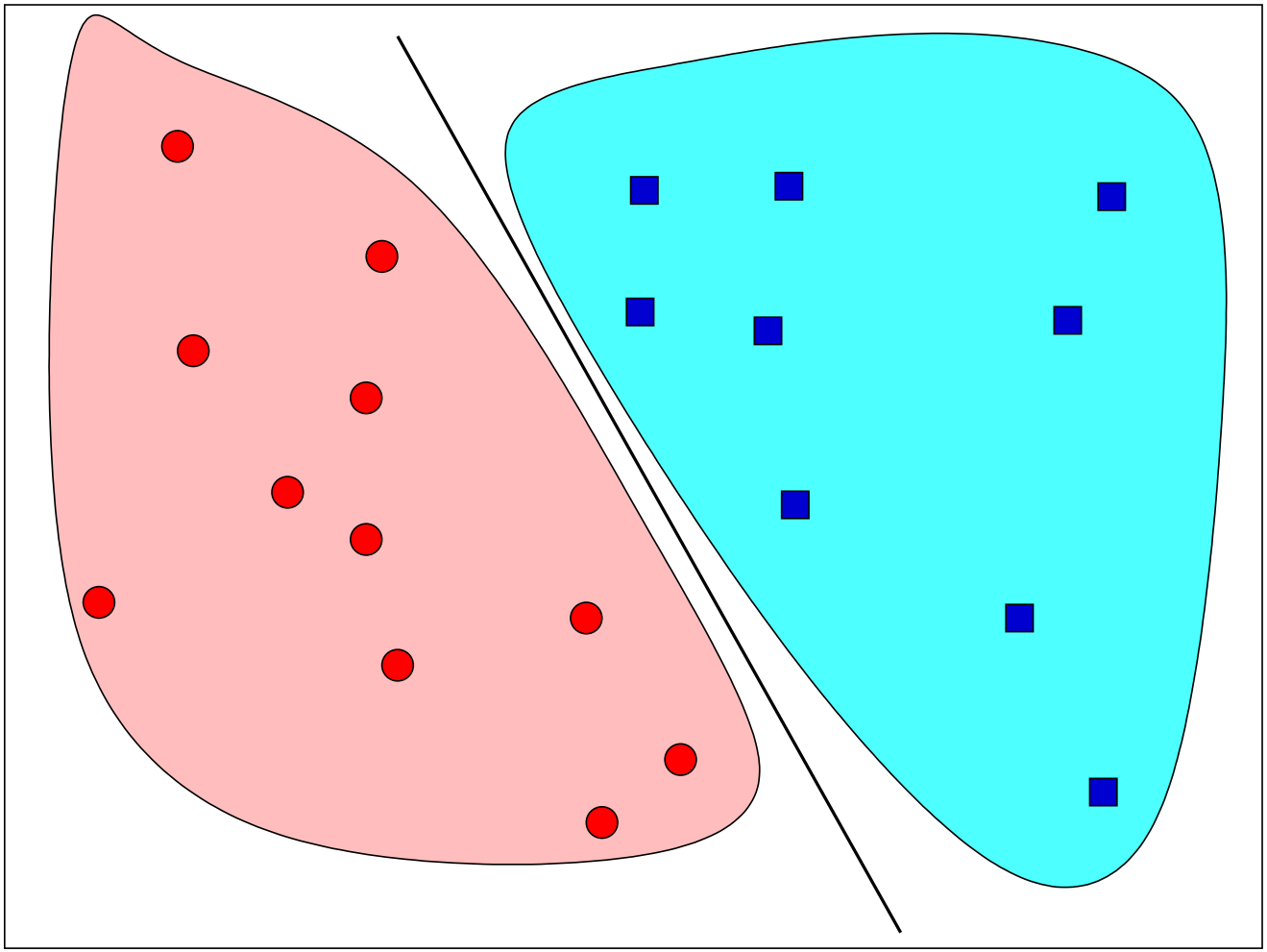
In the objective, the jth loss term is zero when $\text{sign}(a_j^T x - \beta) = y_j$, positive otherwise. A popular function is hinge loss:

$$H(x, \beta) = \frac{1}{m} \sum_{j=1}^{m} \max(1 - y_j(a_j^T x - \beta), 0).$$

Add a regularization term $(\lambda/2)\|x\|_2^2$ for some $\lambda > 0$ to maximize the margin between the classes. And/or $\lambda\|x\|_1$ to sparsify / select features.







Kernel SVM

To enhance "separability" of the data, apply a nonlinear transformation $a_j \to \psi(a_j)$ ("lifting") and do linear classification on $(\psi(a_j), y_j)$: Find (x, β) such that

$$\min_{x,\beta} \frac{1}{m} \sum_{j=1}^{m} \max(1 - y_j(\psi(a_j)^T x - \beta), 0) + \frac{1}{2}\lambda \|x\|_2^2.$$

Can avoid defining ψ explicitly by using instead the dual:

$$\min_{\alpha \in \mathbb{R}^m} \frac{1}{2}\alpha^T Q \alpha - e^T \alpha \;\text{ s.t. }\; 0 \le \alpha \le (1/\lambda)e, \;\; y^T \alpha = 0.$$

where $Q_{k\ell} = y_k y_\ell \psi(a_k)^T \psi(a_\ell)$, $y = (y_1, y_2, \ldots, y_m)^T$, $e = (1, 1, \ldots, 1)^T$.

No need to choose $\psi(\cdot)$ explicitly. Instead choose a kernel K, such that

$$K(a_k, a_\ell) \sim \psi(a_k)^T \psi(a_\ell).$$

[Boser et al., 1992, Cortes and Vapnik, 1995]. "Kernel trick."

SVM Algorithms

Many optimization approaches used for linear and kernel SVM.

- stochastic gradient on the summation (primal) form;
- interior-point [Ferris and Munson, 2000, Fine and Scheinberg, 2001, Gertz and Wright, 2003]
- coordinate descent (SMO) [Platt, 1999]
- dual averaging [Lee and Wright, 2012]
- gradient projection [Serafini et al., 2004, Serafini and Zanni, 2005]

Finite-Sum Structure

$$\min_x f(x) := \sum_{i=1}^{M} f_i(x).$$

Typical application: Empirical Risk Minimization where $f_i(x) := \ell(x; a_i)$

- a_i, $i = 1, 2, \ldots, M$ define the set of M training data;
- x defines the "model";
- ℓ defines the loss — how well the model fits a given data item.

Generalizes linear least-squares: $\ell(x; (a, y)) = x^T a - y$.

Incremental gradient / stochastic gradient are the fundamental algorithms in use here. Enhanced with batching, acceleration, variance reduction (SVRG, SAG, SAGA).

ADMM sometimes also applied to this reformulation:

$$\min_{x, x_1, \ldots, x_M} \sum_{i=1}^{M} f_i(x_i) \quad \text{s.t. } x_i = x, \ i = 1, 2, \ldots, M.$$

Regularization

The optimization problem in ML is usually an empirical proxy for the real problem.

- real problem: defined by expectation over an (unknown) true data distribution;
- empirical problem: defined by expectation over a finite training set, sampled from the true distribution.

Generally, no need to solve the empirical problem exactly. Find ways to incorporate it into a larger strategy, so that the solution generalizes well to the true data distribution.

- explicit regularization — adding terms to the optimization objective;
- use of "tuning sets" and "test sets" to e.g. make good choices of the regularization parameters;
- cross-validation and other statistical criteria for choosing regularization parameters;
- regularization by early stopping of the algorithm;
- use of distributionally robust optimization formulations.

Regularization

Regularization terms added to an empirical objective to avoid overfitting to empirical data; impose structure on a solution.

- $\|\cdot\|_1$ for sparsity / feature selection;
- $\|\cdot\|_2^2$ (Tikhonov) to control model size;
- group-sparse to impose sparsity at a group level;
- TV (sparsity on spatial gradients) for image denoising;
- Nuclear norm $\|\cdot\|_*$ for low-rank matrices.

Sometimes multiple terms are used.

Each term has a nonnegative coefficient λ whose choice is governed by some statistical criterion. Typically need to solve the problem for multiple values of λ, which can affect the choice of optimization algorithm.

Algorithms for Regularized Optimization

$\min_x f(x) + \psi(x)$.

Equivalent to $0 \in \nabla f(x) + \partial\psi(x)$ for convex f, ψ with smooth f.

- proximal gradient [Combettes and Wajs, 2005]
- FISTA: accelerated proximal gradient [Beck and Teboulle, 2009]
- Douglas-Rachford and other operator-splitting methods [1]
- min-max formulations of regularized ERM and resulting algorithms (primal-dual, coordinate descent, reduced variance) [Alacaoglu et al., 2022, Song et al., 2021]
- Coordinate descent with regularization
- mirror descent [Duchi et al., 2010]

[1] Caramanis: `https://www.youtube.com/watch?v=fn3uFc41R60`

Hyperparameter Optimization

We're used to having parameters ("hyperparameters") in optimization algorithms, for sufficient decrease / line search, backtracking, termination, trust region increase / decrease, augmented Lagrangian increase, centering parameter in interior-point, subproblem accuracy, ...

ML adds several more:

- regularization parameters,
- steplength ("learning rate"): value and schedule,
- batch size for finite-sum,
- neural net design parameters,
- ...

In some cases these are critical to reducing solve time to reasonable levels. Often highly problem-dependent.

Nonconvexity in ML

Nonconvexity is a particular focus in modern ML and data science applications.

- nonconvex loss functions e.g. Tukey biweight.
- nonconvex regularization terms in regression e.g. SCAD, MCP.
- nonconvex formulations of low-rank matrix problems: $X = UV^T$, where U and V are tall skinny matrices.
- neural network training.

Weakly convex: subclass of nonconvex where $f(x) + \rho\|x\|^2$ is convex for ρ sufficiently large. Popularized recently by Davis and Drusvyatskiy. (Better complexities available than in for general nonconvex.)

A surprising and very interesting development of recent years is benign or tractable nonconvexity.

Benign Nonconvexity

Despite nonconvexity, useful solutions (even global minima) can be found:

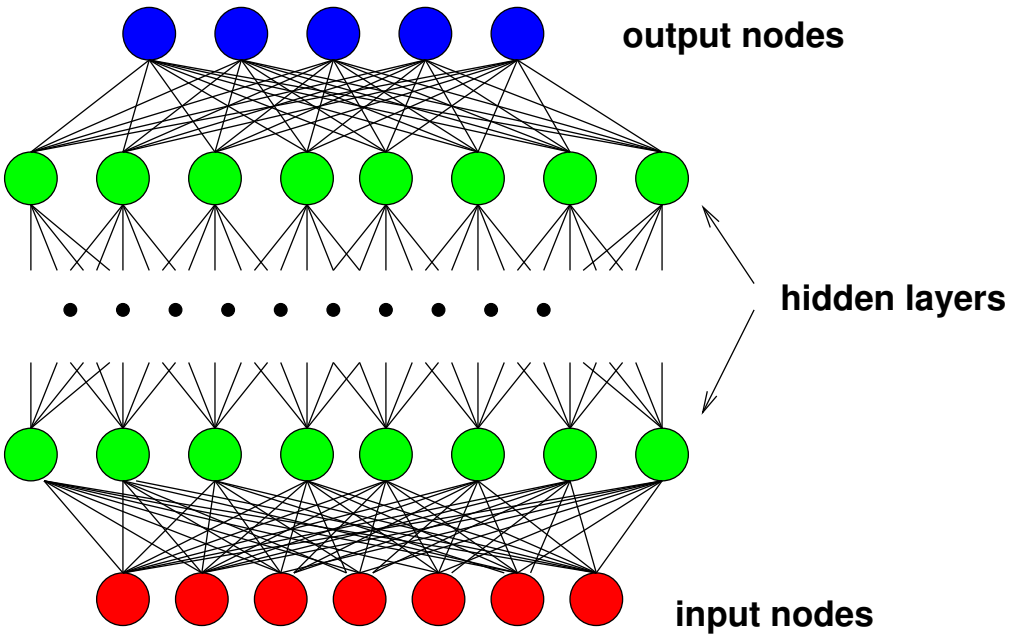
- matrix and tensor problems with explicit low-rank parametrizations;
- dictionary learning;
- phase retrieval;
- overparametrized neural networks;
- AC power flow;
- ...

Ju Sun's excellent page: `https://sunju.org/research/nonconvex/`

Several structures and properties promote benign nonconvexity:

- All local minima are global minima;
- All saddle points are *strict* saddle points, so are easy to escape from (e.g. by detecting negative curvature in the Hessian);
- Smart initialization schemes place x^0 in a neighborhood of a global minimizer.

Neural Network Training



Often used for multiclass classification.

C output nodes, one for each class. Find parameters in the NN such that for input vector a_j, output of node $y_j \in \{1, 2, \ldots, C\}$ dominates the other $C - 1$ outputs.

Maximize cross entropy:

$$\sum_{j=1}^{m} \log \frac{\exp o_{y_j}(a_j, x)}{\sum_{l=1}^{C} \exp o_l(a_j; x)},$$

where $o_l(a_j; x)$ is output of node l for parameters x and input a_j.

Neural Network Training

The objective is nonlinear, nonconvex, nonsmooth.

But it is always optimized using a variant of SGD.

- always with minibatching,
- almost always with "Adam" diagonal scaling [Kingma and Ba, 2015],
- often with a momentum term.

Derivatives calculated by back-propagation ("backprop"), which is reverse-mode automatic differentiation [Griewank and Walther, 2008]. (Nonsmoothness mostly ignored.)

Alternative Objectives for NN Training

Alternative least-squares objective proposed in [Hui and Belkin, 2020]:

$$\sum_{j=1}^{m} \|o(a_j; x) - e_{y_j}\|_2^2$$

where e_{y_j} is the "one-hot vector" in $\mathbb{R}^C$ for class y_j, with 1 in position y_j and zeros elsewhere, and $o(a_j, x)$ is the set of C outputs from the NN.

Experiments in [Hui and Belkin, 2020] show similar performance for cross entropy and least-squares objectives.

Current work of Belkin and Hui with SW indicates that certain combinations of cross-entropy and least-squares can perform even better!

Overparametrization and Benign Nonconvexity

Surprising development of recent years is that for overparametrized NNs,

- a zero-loss solution (global minimizer) is found (if the algorithm is run long enough);
- this solution generalizes well (to unseen data from the same distribution as training data).

See [Belkin, 2021] for a nice review.

These phenomena have been examined from several perspectives:

- "double-descent" explanation of generalization [Belkin et al., 2019]
- landscape analysis of the objective.
- neuro-tangent kernel (NTK): In a nbd of initial values, the problem is nearly linear, so reduces approximately to least-norm solution of underdetermined linear equations.
- mean-field limit and gradient flow
  [Chizat and Bach, 2020, Mei et al., 2018, Wojtowytsch, 2020, E et al., 2020, Ding et al., 2022]

So far, these explanations are for special cases or make assumptions on the NN that are not satisfied in practice.

II. Optimization Algorithms for ML

The demands of ML / Data Science have highlighted needs for certain kinds of algorithms. Most previously known, but in many cases out of fashion and not much studied.

- explosion of interest in first-order methods (usually for problems with Lipschitz gradients). Sublinear convergence is now OK!
- accelerated gradient [Nesterov, 1983]
- stochastic gradient "SGD" [Robbins and Monro, 1951]
  - Adam variant [Kingma and Ba, 2015]: adaptive diagonal scaling
  - variance reduction (e.g. SAG [Schmidt et al., 2016], SVRG [Johnson and Zhang, 2013], both hybrids of full-gradient and stochastic gradient)
  - parallel versions e.g. [Bertsekas and Tsitsiklis, 1989], [Niu et al., 2011]
- coordinate descent (and parallel versions)
- prox-gradient [Combettes and Wajs, 2005, Wright et al., 2009]
  - accelerated: FISTA [Beck and Teboulle, 2009]

Optimization Algorithms for ML, cont'd.

- conditional gradient / Frank-Wolfe [Jaggi, 2013]
- ADMM [Boyd et al., 2011]
- sampled and sketched Newton methods [Byrd et al., 2012], [Bollapragada et al., 2019]
- stochastic quasi-Newton [Byrd et al., 2016]
- bilevel optimization
- primal-dual methods for min-max problems
- algorithms for distributionally robust optimization.

III. Complexity and Convergence

Global complexity analysis of optimization algorithms has become popular.

Not exclusively driven by ML, but dovetails well with ML culture and interests. Many papers written by researchers with ML backgrounds.

- Founded in the oracle complexity concept of [Nemirovski and Yudin, 1983], where the oracle is a unit of information about the function (gradient, unbiased gradient estimate, function+gradient, etc.)
- Other measures of complexity: evaluation, iteration, computational.
- Typically takes the form of a worst-case upper bound on the work required by a specific algorithm to find an ϵ-approx solution to a problem in a given class.
  - expressed in terms of ϵ and other terms that characterize the problem class, e.g. Lipschitz constant of Hessian, lower bound on function, initial value $f(x^0)$, etc.
  - sometimes also expressed in terms of how a measure of optimality decreases with iteration number k.

Complexity

Lower bounds also of interest: *Given a class of algorithms and a class of problems, there is a problem which no algorithm can solve to ϵ-accuracy in less than X amount of work.*

Interesting work to be done in closing gaps between theoretical complexity and the practical behavior of methods.

- "average case" analysis — limited success.
- smoothed analysis [Spielman and Teng, 2004] — hard problems can be converted to easy problems if the data defining a problems is randomly perturbed.
- subdivide the problem class.
- use deeper theoretical insights to explain practical differences between methods with similar theoretical complexities e.g.
  [Lee and Wright, 2018, Wright and Lee, 2020]

Theory and Practice

In some areas, a search for better complexity has yielded better or at least interesting algorithms: e,g. SAG, SAGA, SVRG for SGD.

In others, complexity theory has followed practice e.g. [Niu et al., 2011] and many other parallel methods using SGD and coordinate descent.

In other cases, pursuit of better complexity has led to algorithms that are much slower in practice.

One approach is to take a good practical algorithm and add the bells and whistles needed to equip it with good theory too. e.g.
[Royer et al., 2020, Curtis et al., 2021] for nonconvex smooth unconstrained.

Whither Local Convergence?

There's much less interest in local convergence of Opt-ML algorithms.

- nbd of solution in which local convergence rate cuts in may be small, and we only need an approximate solution.
  - but for some problems we may still need to resolve some *properties* of the solution (e.g. sparsity) even if it's not very accurate.
  - And accurate solutions are back in style in some areas e.g. zero-loss solutions in overparametrized NNs.
- any method that required matrices is too expensive, since dimension is usually large.

Local Convergence

But there are situations where a method with fast local convergence may have advantages. e.g. a skewed convex quadratic function converges in

- $O(\kappa \log \epsilon)$ or $O(\sqrt{\kappa} \log \epsilon)$ iterations of a first-order method.
- Conjugate gradient depends on eigenvalue structure, but likely $\ll n$ iterations.
- One Newton iteration.

More generally if size of domain of local convergence if characterized by $\chi > 0$, complexity of "global phase" depends on $\max(\chi, \epsilon)$ rather than ϵ.

IV. Publications

ML has a conference culture, which changes the way work is done and papers are written.

- Fixed format: short, proofs in appendix, semi-compulsory numerical section.
- Questionable referering standards, due to massive volume of conference submissions.
- Can a conference paper be expanded and published in a journal? Policies differ.
- Journals continue, with page limits and new outlets e.g. SIMODS. Some problems with refereeing times.
- Time to publication date is less critical since outlets like arxiv and Optimization Online are available for preprints. (But there's still value in getting the stamp of approval of journal acceptance.)

Referee resources have not expanded to match the larger number of people and higher rates of publication.

Publications

There's a trend toward longer and more technical papers, with long proofs. e.g. to deal with stochastic gradient, reduced variance, acceleration issues.

The technical wizardry is admirable. But something is lost when papers become too hard to read in detail.

Analysis of a given method is often simplfied later (as in theoretical CS). Simplicity has value! We should consider publishing new, simpler proofs of known results.

V. Parallel Algorithms

Parallel optimization had a boomlet in the mid-late 80s, when many weird and wonderful new parallel architectures appeared (Cray, Sequent Balance, Alliant FX/8, Intel Hypercube, IBM SP, ...).

Partly as an offshoot of parallel linear algebra.

The general optimization paradigms we worked with then were not particularly suitable for parallelism, but there was some work on variable and constraint distribution (Ferris and Mangasarian) and potentially parallel methods for LP (Mangasarian, DeLeone, SW).

Highlight: [Bertsekas and Tsitsiklis, 1989]. Dealt with finite-sum form, synchronous and asynchronous algorithms, etc.

OptML and Parallelism

OptML is a good fit for parallelism:

- finite-sum distributes naturally;
- huge variable space, also distributable (e.g. layers in a NN)
- try different hyperparameter values on different processors.

Moreover parallelism is necessary because training is so compute-intensive.

Much research on techniques to handle relatively slow communication between processors:

- specific communication patterns;
- intermittent communication;
- sparsified communication.

Convergence rates (relative to serial methods) are studied e.g. [Woodworth et al., 2020b, Woodworth et al., 2020a]. Also lower bounds for given setups e.g. [Woodworth et al., 2021, Woodworth, 2021].

VI. Robust Optimization and ML

Robust optimization techniques provide important tools of several kinds for ML.

Adversarial ML. Carefully selected perturbations to data can cause misclassification in wildly unintuitive ways. One solution: seek to correctly classify not just the point a_i but an entire ball of radius ρ centered at a_i:

$$\min_x \frac{1}{m}\sum_{i=1}^m \ell(x; a_i) \quad \to \min_x \frac{1}{m}\sum_{i=1}^m \max_{\|\delta_i\|\le\rho} \ell(x; a_i + \delta_i)$$

(Similar techniques used in the early days of robust optimization.)

Distributional Robustness (DRO).

- View the training data as defining a distribution $\mathbb{Q}_m$;
- Posit that the true data distribution $\mathbb{P}$ is in a ball of radius ϵ around $\mathbb{Q}_m$;
- (Wasserstein metric used to measure distance between distributions)
- Solve a min-max problem to find a robust optimal classifier.

$$\min_x \frac{1}{m}\sum_{i=1}^m \ell(x; a_i) \quad \to \min_x \max_{\mathbb{Q}: d(\mathbb{Q},\mathbb{Q}_m)\le\epsilon} \mathbb{E}_{a\sim\mathbb{Q}}\, \ell(x; a).$$

Much work on properties and tractable formulations of the min-max problem (Kuhn, Harchaoui,...) including relationship to CVaR / superquantile.

Recent paper [Ho-Nguyen and Wright, 2022] showed that for $\ell(x, a)$ being the "zero-one" loss, the DRO formulation involves a ramp-loss functions, which is nonconvex but provably tractable for some $\mathbb{P}$.

Another kind of robustness is to deliberate corruption by an adversary:

- In computation of a gradient $\sum_{i=1}^{m} \nabla f_i(x)$, some fraction ε of the gradients can be incorrect.
- In a parallel computation, some fraction of the processors can be returning false results ("Byzantine").

Provided ε is not too large, and we make some assumptions on the true distribution, there are techniques for removing a subset of evaluations that includes the corrupted evaluations. (Kane, I. Diakonikolas, Rong Ge, ...) They have described a robust gradient descent procedure for minimizing a convex finite-sum function.

Current work: Extend these ideas to nonconvex algorithms.

Summary

The influences of optimization and machine learning / data science / computational statistics on each other since our meeting here 16 years ago have been remarkably deep and widespread.

It has been exciting to do research during this time.

We have many young, incredibly talented researchers working at this intersection, and there is much work left to do.

The future is bright!

When it comes to machine learning and data science, **optimization really ties the room together!**

References I


Alacaoglu, A., Cevher, V., and Wright, S. J. (2022).
On the complexity of a practical primal-dual coordinate method.
arXiv preprint arXiv:2201.07684.


Beck, A. and Teboulle, M. (2009).
A fast iterative shrinkage-threshold algorithm for linear inverse problems.
SIAM Journal on Imaging Sciences, 2(1):183–202.


Belkin, M. (2021).
Fit without fear: The remarkable mathematical phenomenon of deep learning through the prism of interpolation.
Acta Numerica, 30(arXiv:2105.14368):203–248.


Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019).
Reconciling modern machine-learning practice and the classical bias–variance trade-off.
Proceedings of the National Academy of Sciences, 116(32):15849–15854.


Bertsekas, D. P. and Tsitsiklis, J. N. (1989).
Parallel and Distributed Computation: Numerical Methods.
Prentice-Hall, Inc., Englewood Cliffs, New Jersey.


Bollapragada, R., Byrd, R., and Nocedal, J. (2019).
Exact and inexact subsampled Newton methods for optimization.
IMA Journal of Numerical Analysis, 29(2):545–578.

References II


Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992).
A training algorithm for optimal margin classifiers.
In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152.


Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011).
Distributed optimization and statistical learning via the alternating direction methods of multipliers.
Foundations and Trends in Machine Learning, 3(1):1–122.


Byrd, R. H., Chin, G. M., Nocedal, J., and Wu, Y. (2012).
Sample size selection in optimization methods for machine learning.
Mathematical Programming, Series A, 134(1):127–155.


Byrd, R. H., HAnsen, S. L., Nocedal, J., and Singer, Y. (2016).
A stochastic quasi-newton method for large-scale optimization.
SIAM Journal on Optimization, 26(2):1008–1031.


Chizat, L. and Bach, F. (2020).
Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss.
In *Conference on Learning Theory*, pages 1305–1338. PMLR.

References III


Combettes, P. L. and Wajs, V. R. (2005).
Signal recovery by proximal forward-backward splitting.
Multiscale Modeling and Simulation, 4(4):1168–1200.


Cortes, C. and Vapnik, V. N. (1995).
Support-vector networks.
Machine Learning, 20:273–297.


Curtis, F. E., Robinson, D. P., Royer, C. W., and Wright, S. J. (2021).
Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization.
SIAM Journal on Optimization, 31:518–544.


Ding, Z., Chen, S., Li, Q., and Wright, S. J. (2022).
Overparameterization of deep resnet: Zero loss and mean-field analysis.
Journal of Machine Learning Research, 23:1–65.


Duchi, J., Shalev-Shwartz, S., Singer, Y., and Tewari, A. (2010).
Composite objective mirror descent.
Technical Report, University of California-Berkeley.
To appear in Conference on Learning Theory (COLT 2010).


E, W., Ma, C., and Wu, L. (2020).
Machine learning from a continuous viewpoint, I.
Science China Mathematics, 63(11):2233–2266.

References IV


Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004).
Least angle regression.
Annals of Statistics, 32(2):407–499.


Ferris, M. C. and Munson, T. S. (2000).
Complementarity problems in GAMS and the PATH solver.
Journal of Economic Dynamics and Control, 24:165–188.


Ferris, M. C. and Munson, T. S. (2002).
Interior-point methods for massive support vector machines.
SIAM Journal on Optimization, 13(3):783–804.


Fine, S. and Scheinberg, K. (2001).
Efficient svm training using low-rank kernel representations.
Journal of Machine Learning Research, 2:243–264.


Gertz, E. M. and Wright, S. J. (2003).
Object-oriented software for quadratic programming.
ACM Transations on Mathematical Software, 29:58–81.


Griewank, A. and Walther, A. (2008).
Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.
Frontiers in Applied Mathematics. SIAM, Philadelphia, PA, second edition.

References V

Ho-Nguyen, N. and Wright, S. J. (2022).
Adversarial classification via distributional robustness with Wasserstein ambiguity.
Mathematical Programming, Series B, pages 1–37.

Hui, L. and Belkin, M. (2020).
Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks.
In *The Ninth International Conference on Learning Representations (ICLR 2021)*.

Jaggi, M. (2013).
Revisiting Frank-Wolfe: Projection-free sparse convex optimization.
In *Proceedings of the 30th International Conference on Machine Learning*.

Johnson, R. and Zhang, T. (2013).
Accelerating stochastic gradient descent using predictive variance reduction.
In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc.

Kingma, D. P. and Ba, J. (2015).
Adam: A method for stochastic optimization.
In *Proceedings of International Conference on Learning Representations*.

References VI

Lee, C.-p. and Wright, S. J. (2018).
Random permutations fix a worst case for cyclic coordinate descent.
IMA Journal of Numerical Analysis, 39:1246–1275.

Lee, S. and Wright, S. J. (2012).
Manifold identification in dual averaging methods for regularized stochastic online learning.

Journal of Machine Learning Research, 13:1705–1744.

Lu, F., Keles, S., Wright, S. J., and Wahba, G. (2005).
Framework for kernel regularization with application to protein clustering.
Proceedings of the National Academy of Sciences, 102:12332–12337.

Mangasarian, O. L. (1965).
Linear and nonlinear separation of patterns by linear programming.
Operations Research, 13(3):444–452.

Mangasarian, O. L. (1968).
Multisurface method of pattern separation.
IEEE Transactions on Information Theory, 14(6):801–807.

Mei, S., Montanari, A., and Nguyen, P.-M. (2018).
A mean field view of the landscape of two-layer neural networks.
Proceedings of the National Academy of Sciences, 115(33):E7665–E7671.

References VII

Nemirovski, A. S. and Yudin, D. B. (1983).
Problem Complexity and Method Efficiency in Optimization.
John Wiley.

Nesterov, Y. (1983).
A method for unconstrained convex problem with the rate of convergence $O(1/k^2)$.
Doklady AN SSSR, 269:543–547.

Niu, F., Recht, B., Ré, C., and Wright, S. J. (2011).
HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent.
Technical report, University of Wisconsin-Madison.

Platt, J. C. (1999).
Fast training of support vector machines using sequential minimal optimization.
In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA. MIT Press.

Robbins, H. and Monro, S. (1951).
A stochastic approximation method.
Annals of Mathematical Statistics, 22(3).

Royer, C. W., O'Neill, M., and Wright, S. J. (2020).
A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization.
Mathematical Programming, Series A, 180(arXiv:1803.02924):451–488.

References VIII


Schmidt, M., Le Roux, N., and Bach, F. (2016).
Minimizing finite sums with the stochastic average gradient.
Mathematical Programming, 162:83–112.


Serafini, T., Zanghirati, G., and Zanni, L. (2004).
Gradient projection methods for large quadratic programs and applications in training support vector machines.
Optimization Methods and Software, 20(2–3):353–378.


Serafini, T. and Zanni, L. (2005).
On the working set selection in gradient projection-based decomposition techniques for support vector machines.
Optimization Methods and Software, 20:583–596.


Shi, W., Wahba, G., Wright, S. J., Lee, K., Klein, R., and Klein, B. (2008).
LASSO-Patternsearch algorithm with application to opthalmology data.
Statistics and its Interface, 1:137–153.


Song, C., Wright, S. J., and Diakonikolas, J. (2021).
Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums.
In *International Conference on Machine Learning*, pages 9824–9834. PMLR.

References IX


Spielman, D. A. and Teng, S.-H. (2004).
Smoothed analysis of algorithms: Why the simplex method usually takes polynomial time.
Journal of the Association for Computing Machinery, 51(3):385–463.


Tibshirani, R. (1996).
Regression shrinkage and selection via the LASSO.
Journal of the Royal Statistical Society B, 58:267–288.


Wojtowytsch, S. (2020).
On the convergence of gradient descent training for two-layer ReLU-networks in the mean field regime.
arXiv/2005/13530.


Woodworth, B. (2021).
The Minimax Complexity of Distributed Optimization.
PhD thesis, Toyota Technological Institute-Chicago.


Woodworth, B., Patel, K. K., Stich, S. U., Dai, Z., Bullins, B., McMahan, H. B., Shamir, O., and Srebro, N. (2020a).
Is local SGD better than minibatch SGD?
In *ICML*.

References X

Woodworth, B. E., Bullins, B., Shamir, O., and Srebro, N. (2021).
The min-max complexity of distributed stochastic convex optimizayion with intermittent communication.
In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pages 4386–4437.

Woodworth, B. E., Patel, K. K., and Srebro, N. (2020b).
Minibatch vs local sgd for heterogeneous distributed learning.
In *NeurIPS*.

Wright, S. J. and Lee, C.-p. (2020).
Analyzing random permutations for cyclic coordinate descent.
Mathematics of Computation, 89:2217–2248.

Wright, S. J., Nowak, R. D., and Figueiredo, M. A. T. (2009).
Sparse reconstruction by separable approximation.
IEEE Transactions on Signal Processing, 57:2479–2493.