Some Perspectives on Nonconvex Optimization (especially in Learning)

Stephen Wright (UW-Madison)

IPAM, February 2020

IPAM Director of Special Projects Prof. Stan Osher and friend.

Outline

- (Smooth) Nonconvex Unconstrained Optimization: Convergence and Complexity
  - Trust-region algorithm
  - Practical benefits.
- Benign Nonconvexity in Learning
  - Low-Rank Matrix Optimization
  - Distributionally Robust Classification
- Constrained Optimization: Complexity
  - Bound Constraints
  - General Equality Constraints

Collaborators

Clément Royer (Université Paris-Dauphine)
Mike O'Neill (Wisconsin)
Yue Xie (Wisconsin)
Nam Ho-Nguyen (Wisconsin)
Frank Curtis (Lehigh)
Daniel Robinson (Lehigh)

Nonconvex Complexity: Motivation and Context

- Practical algorithms are a traditional area of study in nonconvex optimization.
  - Unconstrained: Newton, quasi-Newton, nonlinear conjugate gradient (CG), ...
  - Constrained: Interior-point / barrier, SQP, augmented Lagrangian, penalty, reduced gradient, ...
- Most classical convergence theory proves two types of results
  - "accumulation points satisfy first-order conditions $\nabla f(x^*) = 0$;"
  - "if the sequence converges to a second-order point ($\nabla f(x^*) = 0$, $\nabla^2 f(x^*) \succ 0$), it converges rapidly."
- Not a lot of work on "global complexity:" Upper bound on the number of iterations (or computational cost) required to find an approximate optimum.

Nonconvex Complexity: Motivation and Context

Meanwhile, complexity results have long been a focus of research in convex optimization.

- polynomial interior-point for LP, convex QP, problems that admit self-concordant barriers (80s-90s).
- momentum methods for nonlinear convex (heavy ball, Nesterov): faster rates than steepest descent (80s, then 2010-)
- subgradient and stochastic subgradient: convergence rates for averaged iterates.

Interest in complexity for nonconvex optimization is more recent. WHY?

- Enhance the theory, possibly the practice too.
- Intense interest from machine learning (ML) — nonconvex applications in matrix optimization with nice properties such as "strict saddle point" and "all local minima are global."
- (Cultural reason: ML people love complexity.)

Unconstrained Nonconvex Complexity

- Optimization literature ($\geq$2006): cubic regularization and trust-region methods with complexity guarantees for 2oN points e.g. [Nesterov and Polyak, 2006, Birgin and Martínez, 2017, Cartis et al., 2011, Curtis et al., 2017a, Curtis et al., 2017b, Martínez and Raydan, 2017]
- Machine learning / optimization researchers since 2014:
  - Adapting accelerated gradient in various ways [Carmon et al., 2017a, Carmon et al., 2017b];
  - Approximate minimization of the cubic approximation [Agarwal et al., 2017];
  - Gradient descent (+ acceleration), with noise injected to escape from saddles [Jin et al., 2017a], [Jin et al., 2017b].

(NOT exhaustive!)

Algorithms for Smooth Nonconvex Optimization

$\min_{x \in \mathbb{R}^n} f(x)$ where f is smooth, nonconvex, and general.

Seek a second-order necessary (2oN) point:

$$\nabla f(x) = 0, \quad \nabla^2 f(x) \succeq 0.$$

Let $\mathcal{D}$ be an open set containing level set $\{x \mid f(x) \leq f(x^0)\}$. Assume

- f is bounded below: $f(x) \geq f_{\text{low}}$ for all x.
- Gradient and Hessian are Lipschitz continuous: For all $y, z \in \mathcal{D}$, have

$$\|\nabla f(y) - \nabla f(z)\| \leq L_g \|y - z\|, \quad \|\nabla^2 f(y) - \nabla^2 f(z)\| \leq L_H \|y - z\|.$$

At any x, have quadratic and cubic upper bounds on f over all $\mathcal{D}$:

$$f(x+p) \leq f(x) + \nabla f(x)^T p + \frac{L_g}{2}\|p\|^2,$$

$$f(x+p) \leq f(x) + \nabla f(x)^T p + \frac{1}{2} p^T \nabla^2 f(x) p + \frac{L_H}{6}\|p\|^3.$$

Approximate 2oN Points & Guarantees

Seek approximate 2oN points satisfying

$$\|\nabla f(x)\| \leq \epsilon_g, \quad \nabla^2 f(x) \succeq -\epsilon_H I,$$

where ϵ_g and ϵ_H are small positive tolerances.

Seek iteration complexities for finding such points. Also seek operation complexities in terms of the number of fundamental operations required. Bound these in terms of ϵ_g and ϵ_H. (Also L_g, L_H, n.)

We take the "fundamental operations" to be

- gradient evaluations, and
- Hessian-vector products,

whose cost is comparable.

Can use computational differentiation ("backprop") or finite differences:

$$\nabla^2 f(x)d \approx \frac{1}{\delta}[\nabla f(x + \delta d) - \nabla f(x)].$$

A Basic Algorithm with Pretty Good Complexity

When L_g and L_H are known, there is an elementary steepest-descent + negative curvature method that finds an approximate 2oN point in $O(\max(\epsilon_g^{-2}, \epsilon_H^{-3}))$ iterations.

For $k = 0, 1, 2, \ldots$:

- If $\|\nabla f(x^k)\| > \epsilon_g$, take a short steepest-descent step:

$$x^{k+1} = x^k - \frac{1}{L_g}\nabla f(x^k).$$

  Use quadratic upper bound to get a decrease of $\geq \epsilon_g^2/(2L_g)$.

- Otherwise, if $\lambda_{\min}(\nabla^2 f(x^k)) < -\epsilon_H$, find direction d^k such that

$$\|d^k\| = 1, \quad (d^k)^T \nabla^2 f(x^k) d^k = \lambda_{\min}^k < -\epsilon_H, \quad \nabla f(x^k)^T d^k \leq 0.$$

  Take a step of length $2|\lambda_{\min}^k|/L_H$ along d^k to get decrease of $\geq \frac{2}{3}\epsilon_H^3/L_H^2$ (using the cubic upper bound).

Iteration Complexity

Because of the lower bound f_{low}, the number of iterations is at most

$$\max\left(2L_g\epsilon_g^{-2}, \frac{3}{2}L_H^2\epsilon_H^{-3}\right)(f(x^0) - f_{\text{low}}).$$

(Our line-search methods use slightly damped Newton steps, which improve from ϵ_g^{-2} to $\epsilon_g^{-3/2}$ without being *too much* more elaborate.)

Cost of each iteration in this scheme includes

- gradient evaluation and
- (sometimes) cost of finding the most negative eigenvalue of $\nabla^2 f(x^k)$.

The second operation may cost $O(n^3)$ (direct implementation on a general dense problem).

Operation Complexity

In fact, for the negative curvature step, don't need the most negative eigenvalue. Need only a direction d such that

$$d^T \nabla^2 f(x^k) d \leq -\tfrac{1}{2}\epsilon_H \|d\|^2.$$

If $\lambda_{\min}(\nabla^2 f(x^k)) \leq -\epsilon_H$, such a d can be computed to probability $1-\delta$ using randomly-started Lanczos iteration at a cost of

$$\min\left\{n, O\left(\sqrt{\frac{L_g}{\epsilon_H}}|\log\delta|\right)\right\}$$

Hessian-vector products.

Hence, operation complexity is a factor of $\epsilon_H^{-1/2}$ worse than iteration complexity.

Line-Search Newton-CG Procedures (**Royer, O'Neill**)

Use of Newton directions improves these worst-case bounds.

Line-search methods in [Royer and Wright, 2018, Royer et al., 2018] use two kind of directions p_k:

- "sufficient" negative curvature for $\nabla^2 f(x_k)$;
- approximate (slightly) damped Newton $-(\nabla^2 f(x_k) + 2\epsilon_H I)^{-1} \nabla f(x_k)$,

and does a backtracking line search along each such direction.

- Monitor the CG procedure for calculating the damped Newton steps to ensure that no more than $O(\epsilon_H^{-1/2})$ steps are taken. (Requires some complicated termination tests.)
- Randomized CG or randomized Lanczos can be used to search for the "sufficient negative curvature" direction for $\nabla^2 f(x_k)$.

Complexity: If $\epsilon_H = \epsilon_g^{1/2}$, the method finds an approximate 2oN point w.h.p. in $\tilde{\mathcal{O}}(\epsilon_g^{-3/2})$ iterations and $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$ operations.

Trust-Region Newton-CG

Trust-region Newton methods for minimizing smooth f solve at k:

$$s^k = \arg \min_{\|s\| \leq \delta_k} m_k(s) := \nabla f(x^k)^T s + \tfrac{1}{2} s^T \nabla^2 f(x^k) s,$$

where δ_k is the trust-region (TR) radius.

Define ratio ρ_k of actual to predicted decrease in f:

$$\rho_k := \frac{f(x^k) - f(x^k + s^k)}{m_k(0) - m_k(s^k)}.$$

- If $\rho_k \geq \eta$ (for some small $\eta > 0$), take step $x^{k+1} = x^k + s^k$
- If $\rho_k \geq \frac{1}{2}$, choose bigger TR for next iteration: $\delta_{k+1} > \delta_k$.
- Otherwise, if $\rho_k < \eta$, reject the step; decrease δ_k, compute new s^k.

Steihaug's method (1980)

Steihaug (1980) applies CG to minimization of model $m_k(s)$.

- Start from $s = 0$;
- If it crosses the TR boundary, stop at the TR boundary and return;
- If negative curvature direction in $\nabla^2 f(x^k)$ is detected, move along that direction to the TR boundary, then return.
- If TR boundary does not interfere, keep iterating to the minimum of m_k. (At most n iterations.)

Properties:

- Popular and practical.
- First step of CG is to the "Cauchy point," which is enough to guarantee overall convergence to a first-order point.
- Each CG step reduces model m_k, and moves further away from 0.
- (No second-order guarantees; method does not move away from a saddle point.)
- No complexity guarantees.

TR Newton-CG: Modifying for Complexity Guarantees (**Royer, Curtis, Robinson**)

[Curtis et al., 2019]: Keep the spirit of Steihaug's method, but modify to enable convergence guarantees.

- Add regularization term to model function:

$$m_k(s) := \nabla f(x^k)^T s + \tfrac{1}{2} s^T \nabla^2 f(x^k) s + \epsilon_H s^T s.$$

- Use the CG method from the line-search method, but modified additionally to stay inside the TR.
- Add a minimum eigenvalue oracle (MEO) to check explicitly for negative curvature (in case CG doesn't find it).

As in the line-search methods, damping with ϵ_H ensures that not too many iterations of CG or MEO are needed to identify directions of "significantly negative curvature."

Minimum Eigenvalue Oracle (MEO)

Inputs: Symmetric $H \in \mathbb{R}^{n \times n}$, scalar M with $\lambda_{\max}(H) \leq M$, and $\epsilon > 0$;
Set parameter $\delta \in [0, 1)$;
Outputs: Estimate λ of $\lambda_{\min}(H)$ such that $\lambda \leq -\epsilon/2$, and vector v with $\|v\| = 1$ such that $v^\top H v = \lambda$ OR certificate that $\lambda_{\min}(H) \geq -\epsilon$.

(If the certificate is output, it is false with probability δ.)

Need MEO, as Modified CG alone may not suffice to identify negative curvature directions, e.g. when $\nabla f(x_k) = 0$ (a possible saddle point).

Can be implemented with randomized Lanczos. Theory from [Kuczyński and Woźniakowski, 1992, Kuczyński and Woźniakowski, 1994] shows that this requires $\tilde{\mathcal{O}}((L_g/\epsilon)^{1/2})$ matrix-vector multiplications with H.

(It can also be implemented with conjugate gradient with a random right-hand side.)

Complexity Results

- CG called at every iteration to compute a step s^k;
- MEO called as check when CG does not return a useful result and gradient $\nabla f(x^k)$ is small.

Complexity results are broadly the same as the line-search methods: To find a point x with

$$\|\nabla f(x)\| \leq \epsilon_g, \quad \nabla^2 f(x) \succeq -\epsilon_g^{1/2} I,$$

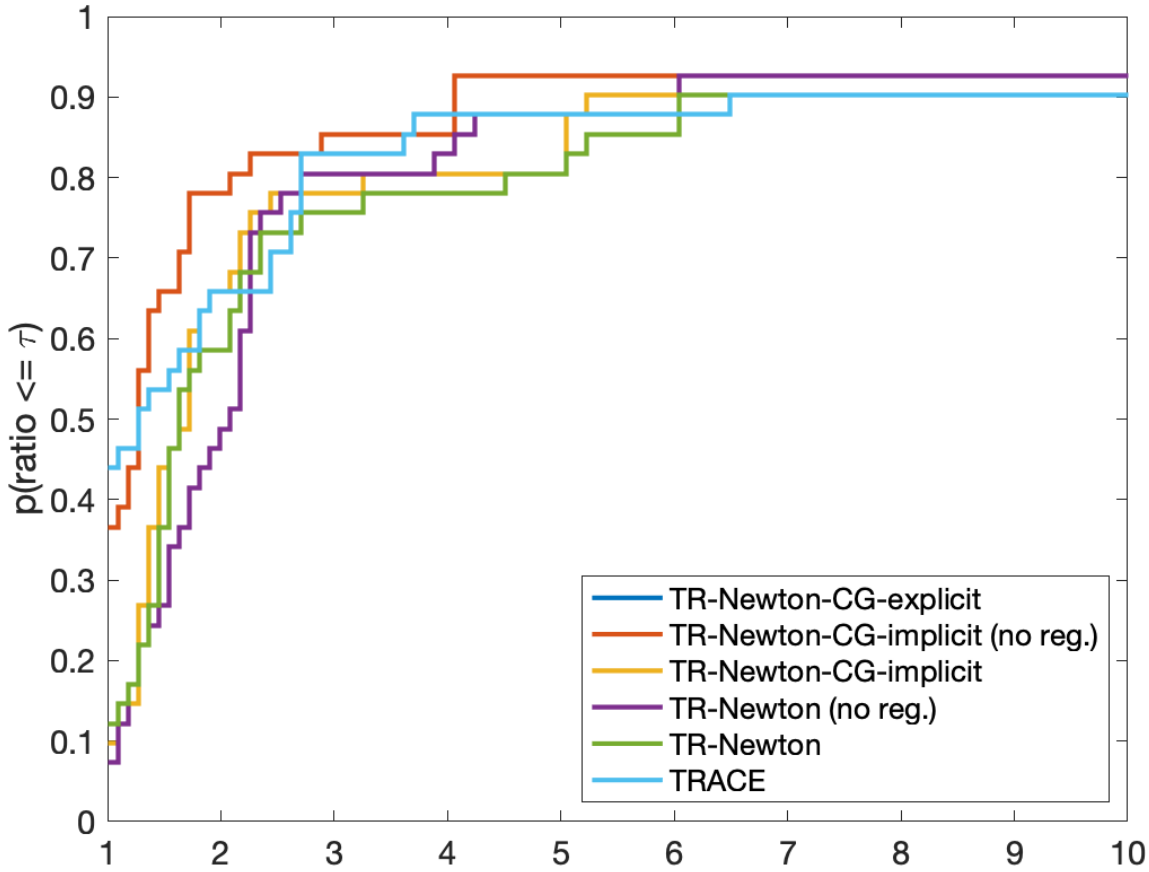
with high probability, need $\tilde{\mathcal{O}}(\epsilon_g^{-3/2})$ iterations and $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$ operations.

Computational Results

Tested several variants of TR-Newton on problems from the CUTEst set with $n > 100$ All variants solve $38/41$ problems within $10^4 n$ iterations.

Inexact (CG) and exact subproblems solution.

Performance profiles show similar # iterations across variants



Nonconvex Optimization in ML

Nonconvexity arises often in model ML, particularly in

- matrix problems with explicit low-rank parametrizations (not convex relaxations);
- phase retrieval;
- neural networks (NNs).

Nonconvexity is often *benign*: Reasonable algorithms find useful solutions, sometimes even global minimizers.

There is some theory to explain this phenomenon in certain cases (see e.g. [Chi et al., 2018] for matrix problems). The statistical properties induce nice properties and structure in the optimization formulation. Examples:

- 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);
- Initialization schemes that place x^0 near a global minimizer.

Example: Matrix Completion (Symmetric)

Matrix Completion with symmetric X:

$$\min_X \frac{1}{2m} \sum_{j=1}^{m} (\mathcal{A}_j(X) - y_j)^2.$$

- When symmetric X has low rank r, write $X = ZZ^T$ where $Z \in \mathbb{R}^{n \times r}$.
- $\mathcal{A}_j(X) = \langle A_j, ZZ^T \rangle$ for some symmetric $A_j \in \mathbb{R}^{n \times n}$.
- Assume that the A_j satisfy a RIP property:

$$(1 - \delta_q)\|X\|_F^2 \leq \frac{1}{m} \sum_{j=1}^{m} \langle A_j, X \rangle^2 \leq (1 + \delta_q)\|X\|_F^2,$$

for all X with rank at most q and some $\delta_q \in (0, 1)$.

Formulation is thus

$$\min_Z h(Z) := \frac{1}{2m} \sum_{j=1}^{m} (\langle A_j, ZZ^T \rangle - y_j)^2.$$

Example: Matrix Completion (Symmetric)

If the properties above hold with $q = 2r$ and $\delta_{2r} \in (0, .1]$, then

- All local minima of F are global;
- All stationary points of F that are not strict have negative curvature in $\nabla^2 F(Z)$.

[Bhojanapalli et al., 2016]

Smart initialization: The matrix

$$Y := \frac{1}{m} \sum_{j=1}^{m} y_j A_j$$

is close to the solution if RIP properties are satisfied for $q = 2r$. Steepest descent on F can converge from such a starting point.

Low-Rank Matrix Opt: $\log \epsilon$ Complexity (**M. O'Neill**)

[Zhu et al., 2017] describe a class of low-rank matrix optimization problems that generically has a strict saddle property:

> *Any stationary point* $\nabla f(x) = 0$ *is either a local solution* ($\nabla^2 f(x) \succeq 0$) *or else has* $\lambda_{\min}(\nabla^2 f(x)) \leq -\sigma$, *for some* $\sigma > 0$.

Since σ is constant — does not depend on whatever tolerance ϵ we choose for approximate minimizers — we can escape from such points at a cost independent of ϵ, thus get convergence rates that depend much more weakly on ϵ than in the general case.

Form of the problem considered by [Zhu et al., 2017]:

$$\min_W F(W) := f(UV^T), \text{where } W = \begin{bmatrix} U \\ V \end{bmatrix} \in \mathbb{R}^{(m+n)\times r}.$$

Remove scaling ambiguity in the W corresponding to X by regularizing:

$$G(W) := F(W) + \frac{1}{2}\|U^T U - V^T V\|_F^2.$$

Further Assumptions and Construction

Assume: f has a critical point X^* ($\nabla f(X^*) = 0$) with rank r.

Assume: uniform restricted strong convexity condition for f: there are positive constants a and b such that

$$a\|T\|_F^2 \leq [\nabla^2 f(X)](T,T) \leq b\|T\|_F^2$$

where $\mathrm{rank}(X) \leq 2r$ and $\mathrm{rank}(T) \leq 4r$. Also need

$$\frac{b-a}{b+a} \leq \frac{\sigma_r(X^*)^{3/2}}{\|X^*\|_F \|X^*\|^{1/2}}.$$

The space of W is covered by three regions:

- $\mathcal{R}_1$: $\mathrm{dist}(W, W^*) \leq \sigma_r^{1/2}(X^*)$: close to X^*;
- $\mathcal{R}_2$: $\lambda_{\min}(\nabla^2(G(W))) \leq -\sigma_r(X^*)$: large negative curvature;
- $\mathcal{R}_3$: $\|\nabla G(W)\|_F \geq \min(\sigma_r(X^*)^{3/2}, \|W\|^3, \|WW^T\|_F^{3/2})$ (large gradient).

Sketch of the Algorithm

We can't just apply the methods for general f and automatically get $\log \epsilon$ complexity. (If we knew $\sigma_r(X^*)$, we could improve the dependence on ϵ.)

We propose a specialized method that does not require knowledge of $\sigma_r(X^*)$ is unknown — but maintains a proxy γ_k.

- If gradient $\nabla G(W^k)$ is large (relative to $\gamma_k^{3/2}$), take a steepest descent step (with backtracking).
- Else test for negative curvature.
  - If not negative curvature, we might be close to X^*: Try inner loop: gradient steps and test for linear convergence.
    - ⋆ If not linear, conclude that γ_k overestimates $\sigma_r(X^*)$, so halve it for next iteration.
    - ⋆ Else Converged!
  - Else do backtracking line search along the negative curvature direction.

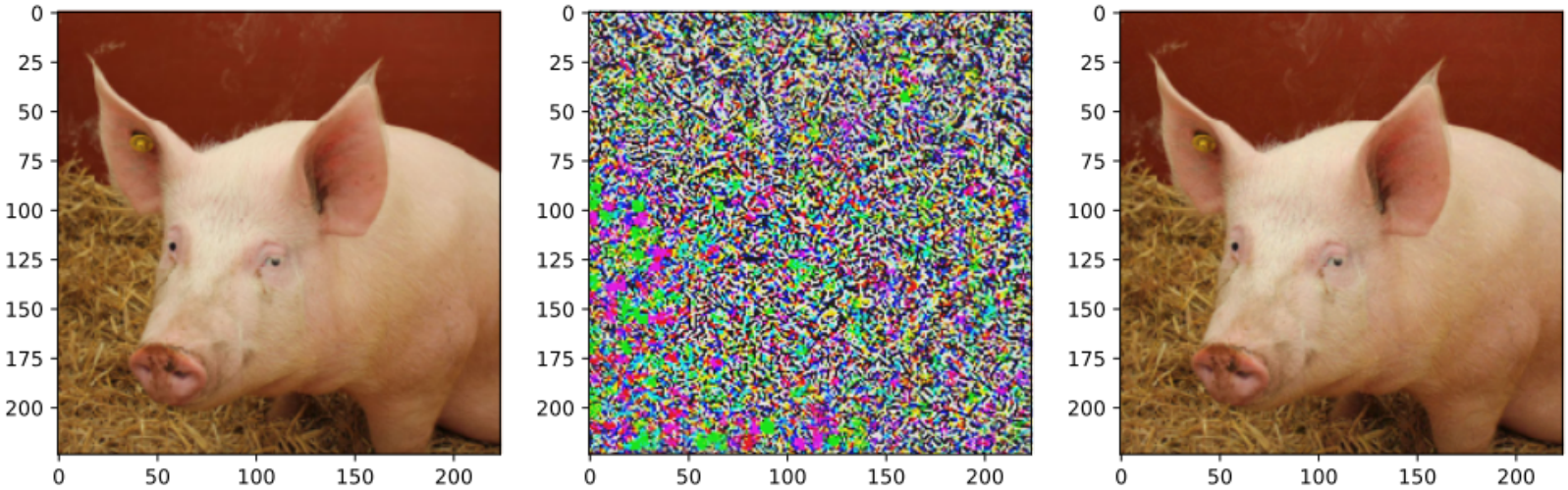
Complexity

Dependence on accuracy tolerance ϵ is only through a log term.

- Inner loop requires $O(\log \epsilon)$ steps to achieve an ϵ-accurate second-order point.
- Main algorithm: number of iterations is independent of ϵ (related instead $\sigma_r(X^*)^{-3}$.)

Adversarial Classification (DRO) (**Nam-Ho Nguyen**)

Problem in image classification: small perturbations of images can change the classification![1]



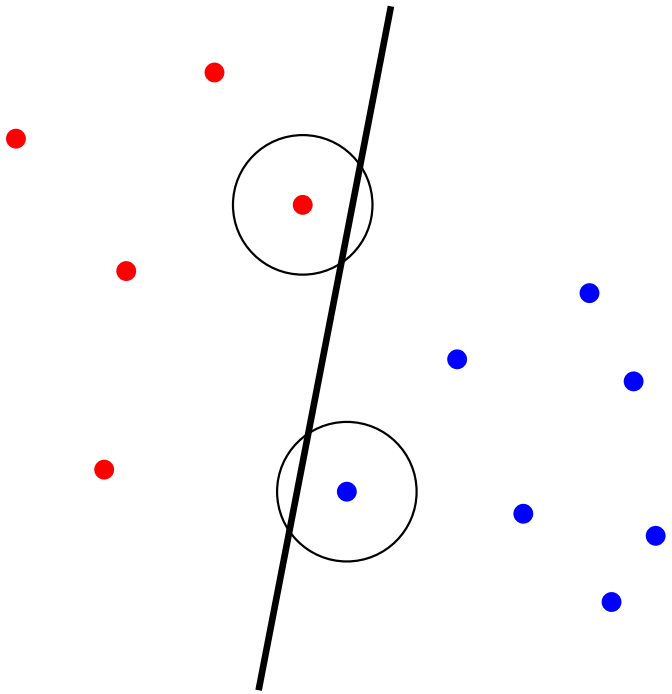
Left: image of pig, classified correctly.
Right: incorrectly classified (wombat) identical pig obtained by adding visually imperceptible noise **(middle)**.

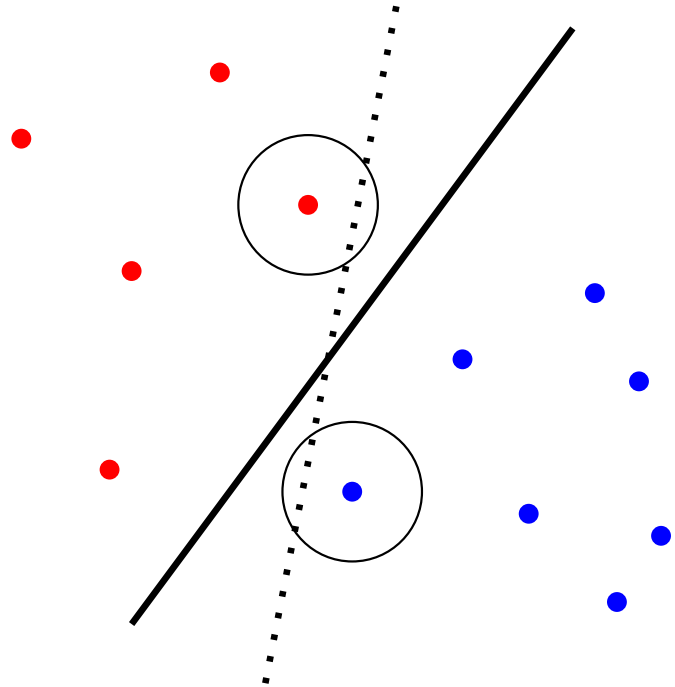
[1] `https://adversarial-ml-tutorial.org/introduction/`

Adversarial Classification

This phenomenon arises because the decision boundary is too close to the data points.

Ideally, want a decision boundary that will give the same classification even if points are not exactly at the stated location.





Adversarial Binary Classification

We have points $(x, y) \in X \times \{\pm 1\}$ distributed according to P. We want a function $f : X \to \mathbb{R}$ such that $\text{sign}(f(x)) = y$.

- (x, y) is misclassified $\iff$ $yf(x) \leq 0$.
- Prefer f such that $\mathbb{P}_{(x,y)\sim P}[yf(x) \leq 0]$ is small.

Define distance to boundary:

$$\text{dist}(x, y, f) := \min_{\Delta} \{\|\Delta\| : yf(x + \Delta) \leq 0\} \quad (\Delta = 0 \text{ if already misclassified})$$

Three measures of robustness for f (for fixed $\epsilon > 0$):

- $\boxed{\mathbb{P}_{(x,y)\sim P}[\text{dist}(x, y, f) \leq \epsilon]}$ perturbation robustness.
  - (small is better)
- $\boxed{\mathbb{E}_{(x,y)\sim P}[\text{dist}(x, y, f)]}$ expected distance to boundary
  - (large is better)
- $\boxed{\max_{d(Q,P)\leq\epsilon} \mathbb{P}_{(x,y)\sim Q}[\text{dist}(x, y, f) = 0]}$ distributional robustness
  - (small is better)

Distributional Robustness

We don't know P, but have i.i.d. samples $(x_i, y_i) \sim P$, $i \in [n]$. Let $\hat{\mathbf{P}}_\mathbf{n}$ be the empirical distribution.

We want a classifier with **optimal distributional robustness**:

$$\min_{f \in \mathcal{F}} \max_{d(Q, \hat{\mathbf{P}}_\mathbf{n}) \leq \epsilon} \mathbb{P}_{(x,y)\sim Q}[\text{dist}(x, y, f) = 0].$$

- Expected distance from boundary has been shown to have undesirable properties [Fawzi et al., 2018].
- Distributional robustness has attractive out-of-sample guarantees.
  - For n sufficiently large, $d(P, \hat{P}_n) \leq \epsilon$ with high probability.
- We use **Wasserstein distances** for adversarial classification:

$$d_W(Q, P) = \min_{\Pi} \left\{ \mathbb{E}_{(x,x')\sim\Pi}[\|x - x'\|] : \Pi \text{ has marginals } P_X, Q_X \right\}.$$

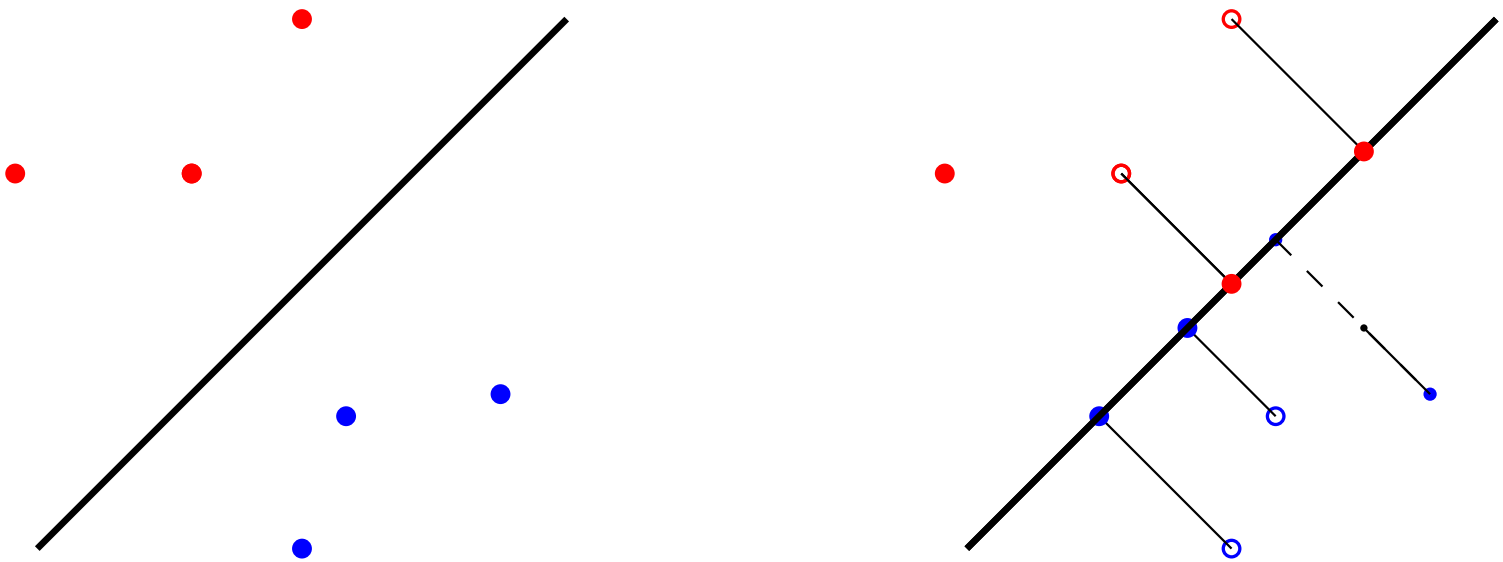
Recently popular in data-driven optimization [Mohajerin Esfahani and Kuhn, 2018, Gao and Kleywegt, 2016, Blanchet and Murthy, 2019].

Wasserstein Worst-Case Distributions

Fix a classifier $f \in \mathcal{F}$. Consider worst-case distribution [Chen et al., 2018]:

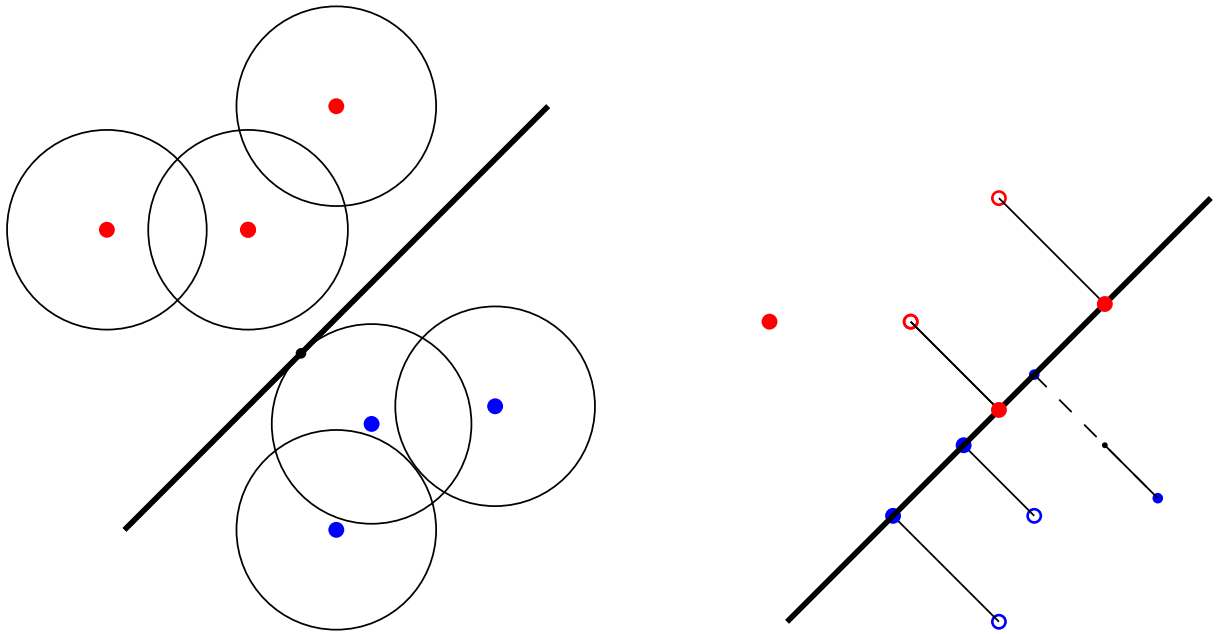
$$Q^* = \arg \max_{d_W(Q,\hat{P}_n) \leq \epsilon} \mathbb{P}_{(x,y)\sim Q}[\text{dist}(x, y, f) = 0].$$

- Q^* aims to transport as many points (x_i, y_i) to the misclassification set $\{(x, y) : \text{dist}(x, y, f) = 0\}$ as possible.
- For 1-norm, the sum of transported distances must be $\leq \epsilon n$.
- If it cannot transport a whole point, it can 'split' a point.



Wasserstein vs Perturbation Robustness

- Consider a fixed $f \in \mathcal{F}$. Perturbation robustness: tries to perturb x_i by Δ_i, where $\|\Delta_i\| \leq \epsilon$, to misclassify $y_i f(x_i + \Delta_i) \leq 0$.
- Similar to the Wasserstein worst-case distribution, the sum of perturbed distances is $\leq \epsilon n$.



- However, the distance from a single point x_i to its perturbation $x_i + \Delta$ is at most ϵ, whereas the Wasserstein worst-case distribution can perturb a single point x_i by distance up to ϵn.

Optimizing for Wasserstein Distributional Robustness

From the structure of the worst-case distribution:

$$\max_{d_W(Q,\hat{P}_n)\le\epsilon} \mathbb{P}_{(x,y)\sim Q}[\mathrm{dist}(x,y,f)=0] = \min_{t\ge 0}\left\{\epsilon t + \frac{1}{n}\sum_{i\in[n]} \max\{0, 1-\mathrm{dist}(x_i,y_i,f)t\}\right\}$$

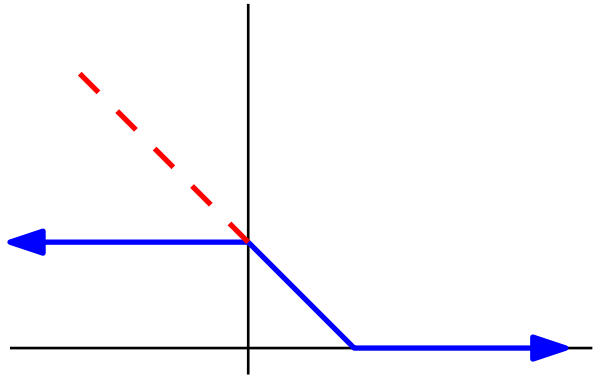
Nice representations of $\mathrm{dist}(x,y,f)$ exist for certain classes $\mathcal{F}$ e.g. linear:

$$f(x)=\langle w,x\rangle \implies \mathrm{dist}(x,y,f)=\max\left\{0,\frac{y\langle w,x\rangle}{\|w\|_*}\right\}.$$

Transforming $\bar{w}=tw/\|w\|_*$, we have

$$\min_{w}\max_{d_W(Q,\hat{P}_n)\le\epsilon} \mathbb{P}[y\langle w,x\rangle\le 0] = \min_{\bar{w}}\left\{\epsilon\|\bar{w}\|_* + \frac{1}{n}\sum_{i\in[n]} L_R(y_i\langle\bar{w},x_i\rangle)\right\}.$$

Ramp Loss L_R is Nonconvex!



This is actually pretty well known as a classification loss, as an alternative to the hinge loss traditionally used in SVM because of outliers (lessens impact from severely misclassified points).

We tried finding global solutions using integer-programming formulations. But this can handle only very limited problem sizes so far ($n < 75$).

An "outer approximation" strategy is even slower.

But for reasonable data, based on our (limited) experience, this seems to be one of those benign nonconvex problems.

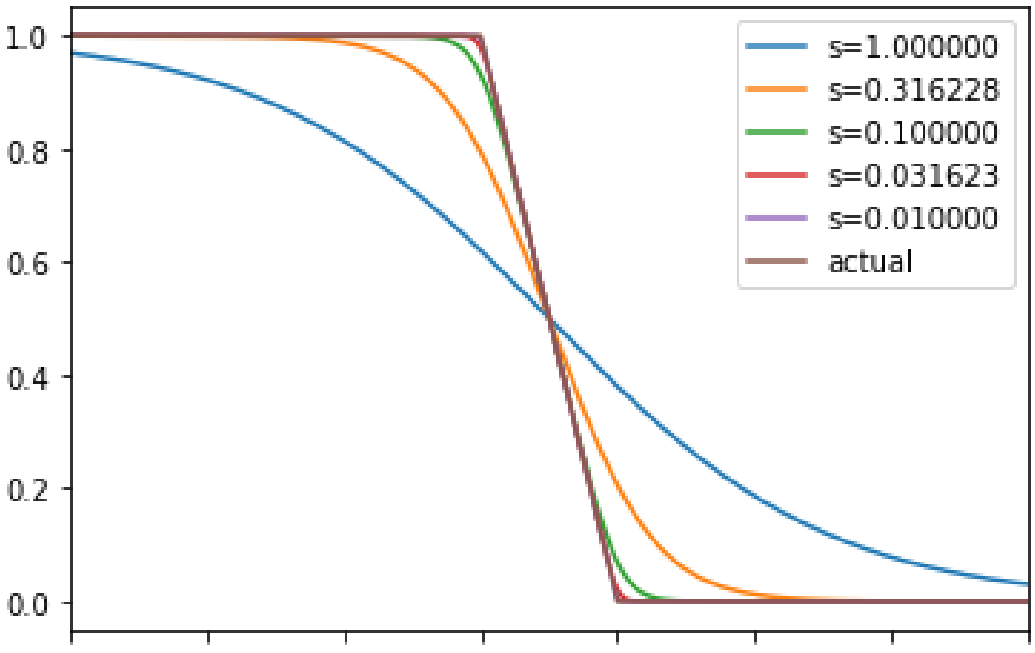
Smoothing the Ramp Loss Function

Ramp loss function is

$$\begin{cases} 1 & \text{if } r < 0 \\ 1 - r & \text{if } r \in [0, 1] \\ 0 & \text{if } r > 1 \end{cases}$$

Approximate it with the smooth function $\psi(r)$ defined by

$$\psi_\sigma(r) := \sigma \log \left(\frac{\exp(1/\sigma) + \exp(r/\sigma)}{1 + \exp(r/\sigma)} \right).$$



Classification

The classification problem with smoothed ramp loss can be formulated as

$$\min_w \epsilon\|w\|_2 + \frac{1}{n}\sum_{i=1}^{n} \psi_\sigma(y_i\langle w, x_i\rangle),$$

which is equivalent for some $\bar{\epsilon} > 0$ to

$$\min_w \frac{1}{2}\bar{\epsilon}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \psi_\sigma(y_i\langle w, x_i\rangle),$$

Solve with standard methods for smooth unconstrained optimization: e.g. nonlinear conjugate gradient, L-BFGS.

Separable Data

- Choose n training points in $x_i \in [-10, 10]^d$, uniformly distributed.
- Set $y_i = \text{sign}([x_i]_1)$ (sign of first component of x_i).
- Thus $\bar{w} = (1, 0, 0, \ldots, 0)^T$ defines a separating hyperplane.
- Set $\sigma = .1$ and minimize from random starting points w^0.

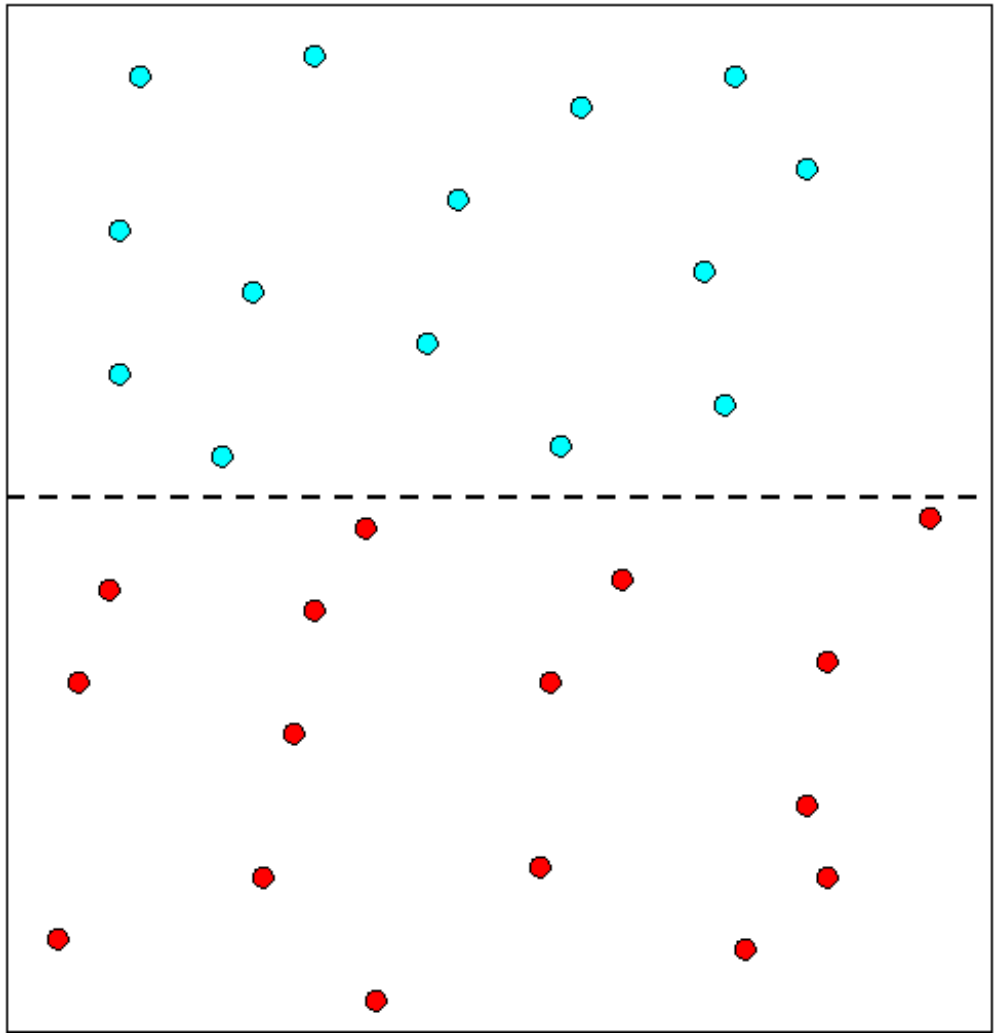
Tried $d = 10$ and $n = 10^2 - 10^5$. Solve in a few seconds.

For $d = 10$ and $n \geq 500$,

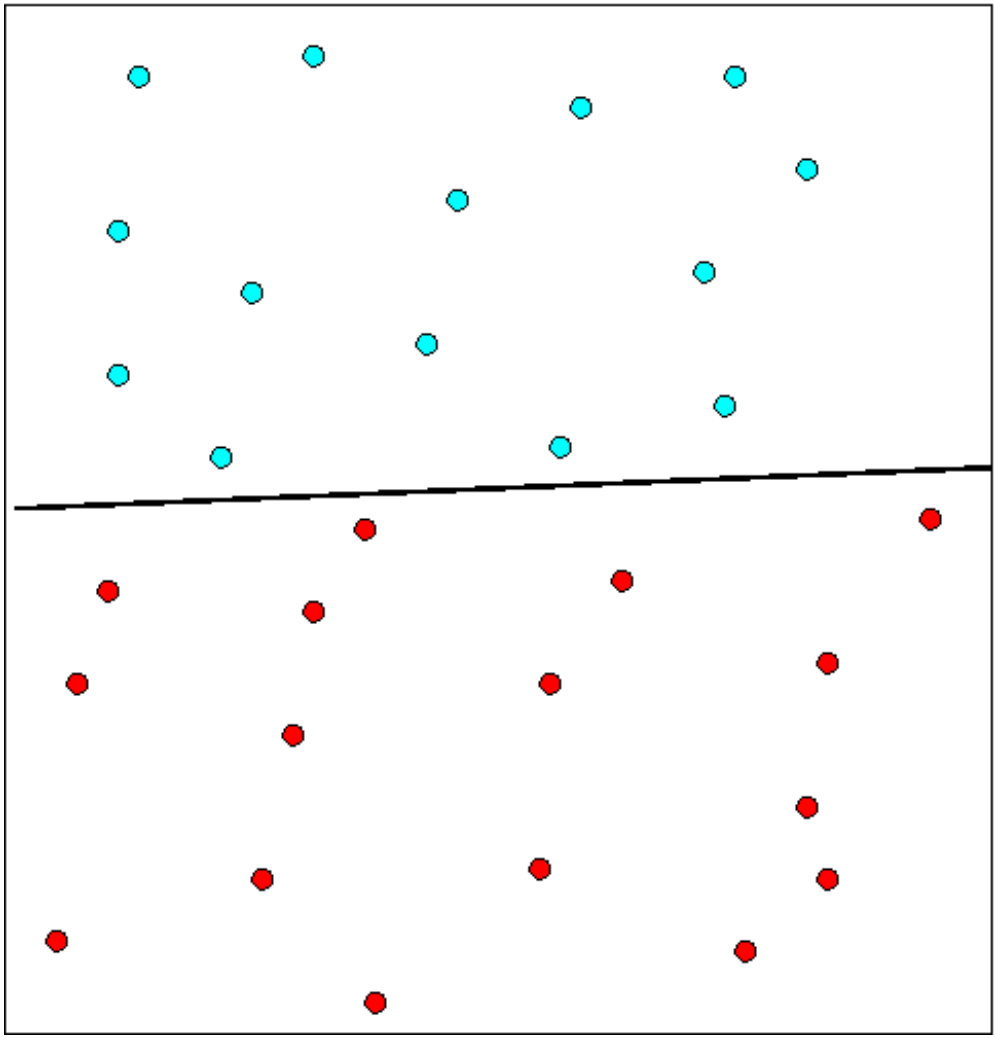
- finds an optimal w^* close (but not identical) to $\bar{w}$.
- Converges to this w^* from any random start.

(For $n = 100$ and $n = 200$, finds local solutions. Here the data is too sparse relative to the dimension $d = 10$ to specify an "obvious" separating plane.)

Random Data (Artificial)



Random Data (Artificial)



Non-Separable Data

Same setup as before, but flip some of the labels.

For 10% of labels flipped, still finds a solution w^* close to $\bar{w}$ from any random start.

For 20% of labels flipped, occasionally finds a local solution.

Benign Nonconvexity

Why is the solution "easy" to find, and so robust to label flipping?

Get some clues by looking more closely at the toy example above, where $y_i = \text{sign}([x_i]_1)$ and $w^* \approx (1, 0, 0, \ldots, 0)^T$. Loss function is

$$F_{\bar{\epsilon}}(w) := \frac{1}{2}\bar{\epsilon}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \psi_\sigma(y_i\langle w, x_i\rangle).$$

From $\nabla F_{\bar{\epsilon}}(w^*) = 0$, we have

$$w^* = -\frac{1}{n\bar{\epsilon}}\sum_{i\in[n]} \psi'_\sigma(y_i\langle w, x_i\rangle)y_i x_i.$$

Note that $\psi'_\sigma(\cdot) < 0$ for all arguments.

For separable data, since $y_i[x_i]_1 > 0$ for all i, *the first component of all terms in the summation above has the same sign*, so we expect w_1^* to be large and positive.

For $j = 2, 3, \ldots, d$, the sign of $y_i[x_i]_j$ is arbitrary, so by cancellation we expect each w_j^* to be much smaller in magnitude than w_1^*.

Benign Nonconvexity

This logic suggests that $w^* \approx (1, 0, 0, \ldots, 0)^T$ would be the only credible stationary point! So any algorithm that finds even a stationary point should converge to this solution.

No need to even consider second-order optimality, negative curvature directions, etc.

The same reasoning explains why $(1, 0, 0, \ldots, 0)^T$ is approximately stationary even when labels are flipped. There is still a "bias" in the summation over the first components $y_i[x_i]_1$ — most have the same sign — while cancellation happens for the remaining components.

Complexity for Constrained Nonconvex Optimization

$\min f(x)$ subject to $x \in S$, where S is some closed set.
Many algorithms with $\mathcal{O}\left(\epsilon_g^{-3/2}\right)$ iteration complexity for some approximate optimality condition when S is "simple" (e.g. projection onto S can be computed efficiently).

- Methods extending cubic regularization [Cartis et al., 2012, Cartis et al., 2015, Cartis et al., 2018].
- Active set method [Birgin and Martínez, 2018].
- Interior point/barrier method [Haeser et al., 2018].

No dimension-free operational complexity results for these algorithms.

Nonnegativity Constraints (**M. O'Neill**)

min $f(x)$ subject to $x \geq 0$: The simplest inequality constrained problem.

First-order conditions: $0 \leq x^* \perp \nabla f(x^*) \geq 0$.

Less tersely: Can partition $\{1, 2, \ldots, n\} = \mathcal{A} \cup \mathcal{I} \cup \mathcal{D}$ such that

- $x_i^* = 0$, $\nabla_i f(x^*) > 0$ for $i \in \mathcal{A}$ (active);
- $x_i^* > 0$, $\nabla_i f(x^*) = 0$ for $i \in \mathcal{I}$ (inactive);
- $x_i^* = 0$, $\nabla_i f(x^*) = 0$ for $i \in \mathcal{D}$ (degenerate).

Q. Is there a generalization of line-search Newton-CG that converges to approximate second-order necessary points with complexity guarantees?

A. Depends what you mean by "second-order necessary (2oN) conditions."

The strongest 2oN conditions are that $v^T \nabla^2 f(x^*) v \geq 0$ for v such that

$$\mathcal{S}_2 = \{v_i = 0,\ i \in \mathcal{A};\quad v_i \geq 0,\ i \in \mathcal{D}\}.$$

But it can be NP-hard to check this condition.

Second-Order Necessary

$f(x) := x^T Q x$ for symmetric Q. Satisfies first-order conditions with $\mathcal{D} = \{1, 2, \ldots, n\}$. But in this case 2oN conditions = copositivity of Q.

This is a worst case — could still be checkable if $|\mathcal{D}|$ is not too large.

The standard "cop-out" (dating to at least 1990) is to aim for a weaker form of 2oN conditions:

$$[\nabla^2 f(x^*)]_{\mathcal{I}\mathcal{I}} \succeq 0.$$

Define $\bar{x} = \min(x, \mathbf{1})$ and $\bar{X} = \text{diag}(\bar{x})$.

We work with the following approximate 2oN conditions (similar to [Haeser et al., 2018], except that they use X instead of $\bar{X}$).

$$x > 0, \quad \nabla f(x) > -\epsilon_g \mathbf{1}, \quad \|\bar{X} \nabla f(x)\|_\infty \leq \epsilon_g,$$

$$\bar{X} \nabla^2 f(x) \bar{X} \succeq -\epsilon_H I.$$

Log-Barrier Approximation

We reduce the bound-constrained problem to unconstrained minimization of the log barrier function:

$$\phi_\mu(x) := f(x) - \mu \sum_{i=1}^{n} \log(x_i)$$

for some $\mu > 0$. Only defined on the interior of the set $x \geq 0$.

We minimize this for a single (small) value of μ, chosen so that near-optimal second-order points for ϕ_μ satisfy the approximate second-order conditions for the bound-constrained problem.

Plan: Use our Newton-CG approach — modified to ensure positivity of all iterates x^k) — to minimize this function efficiently.

[O'Neill and Wright, 2019]

Modifying Newton-CG for the Log-Barrier Function

Gradient and Hessian of the log-barrier function are:

$$\nabla \phi_\mu(x) = \nabla f(x) - \mu X^{-1} e \quad \text{and} \quad \nabla^2 \phi_\mu(x) = \nabla^2 f(x) + \mu X^{-2}.$$

Modify Newton-CG as follows:

- Set $\mu = \frac{1}{4}\epsilon_g$ and $\epsilon_H = \sqrt{\epsilon_g}$.‘
- Precondition / scale the Newton equations with the diagonal $\bar{X}$.
- Keep iterates interior to the nonnegative orthant with a "fraction-to-the-boundary" rule: $x^k + d^k \geq (1-\beta)x^k$ where $\beta \in [\epsilon_H, 1)$.
- Decrease in terms of optimality conditions: Add extra termination test to Modified CG ($\|r^j\|_\infty \leq \bar{\zeta}\mu$).

Log-Barrier Newton-CG

if Not first-order optimal **then**
  Call Modified CG with $H = \bar{X}_k \nabla^2 \phi_\mu(x^k) \bar{X}_k$ and $g = \bar{X}_k \nabla \phi_\mu(x^k)$;
  if step type = "negative curvature" **then**
    Scale d to stay interior and flip sign to get d^k;
  else {step type is "damped Newton"}
    Scale d to stay interior to get d^k
  end if
else
  Call MEO with $H = \bar{X}_k \nabla^2 f(x^k) \bar{X}_k$ to output v;
  if MEO certifies that $\lambda_{\min}(\bar{X}_k \nabla^2 f(x^k) \bar{X}_k) \geq -\epsilon_H$ **then**
    Terminate;
  else {direction of sufficient negative curvature found}
    Scale v to stay interior and flip sign to get d^k;
  end if
end if
Line Search: Require $\phi_\mu(x^k + \alpha_k \bar{X}_k d^k) < \phi_\mu(x^k) - \frac{\eta}{6} \alpha_k^3 \|d^k\|^3$;
$x^{k+1} \leftarrow x^k + \alpha_k \bar{X}_k d^k$;

Sufficient Decrease in Log-Barrier Function

By our $1-\beta$ "fraction-to-the-boundary" rule, there is a cubic upper bound on the logarthmic portion of ϕ_μ:

$$
\begin{aligned}
&-\sum_{i=1}^{n} \log(x_i + \bar{x}_i d_i) + \sum_{i=1}^{n} \log(x_i) \\
&\leq -e^T X^{-1} \bar{X} d + \frac{1}{2} d^T \bar{X} X^{-2} \bar{X} d + \frac{2-\beta}{6(1-\beta)^2} \|d\|^3.
\end{aligned}
$$

Combining with the cubic upper bound of f gets at least $\mathcal{O}\left(\epsilon_g^{3/2}\right)$ decrease at each step.

Theorem [O'Neill and Wright, 2019]

Assume f smooth and bounded below, and we use Log-Barrier Newton-CG to seek an approximate second-order point.

- Iteration complexity is $\bar{K}_2 = \tilde{\mathcal{O}}\left(n\epsilon_g^{-1/2} + \epsilon_g^{-3/2}\right)$ with probability at least $(1-\delta)^{\bar{K}_2}$.
- Operation complexity is $\tilde{\mathcal{O}}(n\epsilon_g^{-3/4} + \epsilon_g^{-7/4})$ for large n and $\tilde{\mathcal{O}}(n\epsilon_g^{-3/2})$ for smaller n.

- The "n" term seems to be an unavoidable consequence of using the log-barrier function. Best previous result is $\tilde{\mathcal{O}}(n\epsilon^{-3/2})$ — we do better in the "large n" case.
- If we assume a priori that $\{x^k\}$ is bounded, get $\tilde{\mathcal{O}}(\epsilon_g^{-7/4})$ operation complexity.
- Complexities to get an approximate first-order point are the same, but without the possibility of failure.

Equality Constraints (**Yue Xie**)

$$\min f(x) \quad \text{s.t. } c(x) = 0,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ as before, and $c : \mathbb{R}^n \to \mathbb{R}^m$ is a vector function of equality constraints. f and c are smooth.

For some $\epsilon > 0$, ϵ-1o solution requires:

$$\|\nabla f(x) + \nabla c(x)\lambda\| \leq \epsilon, \quad \|c(x)\| \leq \epsilon.$$

ϵ-2o solution requires also:

$$d^T \left(\nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(x) \right) d \geq -\epsilon \|d\|^2,$$

for any $d \in \mathbb{R}^n$ such that $\nabla c(x)^T d = 0$.

Proximal Augmented Lagrangian (PAL) algorithm

The augmented Lagrangian is

$$\mathcal{L}_\rho(x, \lambda) \triangleq f(x) + \lambda^T c(x) + \frac{\rho}{2}\|c(x)\|^2,$$

where $\rho > 0$ and $\lambda \triangleq (\lambda_1, \ldots, \lambda_m)^T$.

PAL Algorithm:

0. Initialize x_0, λ_0 and $\rho > 0$, $\beta > 0$; Set $k := 0$;
1. Update x_k: Find approximate solution x_{k+1} to

$$\text{argmin } \mathcal{L}_\rho(x, \lambda_k) + \frac{\beta}{2}\|x - x_k\|^2;$$

2. Update λ_k: $\lambda_{k+1} := \lambda_k + \rho c(x_{k+1})$;
3. If termination criterion is satisfied, STOP; otherwise, $k := k + 1$ and return to Step 1.

[Xie and Wright, 2019]

Complexities and Assumptions

PAL involves two levels of iteration: the outer iteration, and the inner iterations to solve the nonconvex unconstrained subproblem.

Three types of complexity:

- Outer iteration complexity
- Total iteration complexity: total number of iterations of the inner loop
- Operation complexity: bound on number of first-derivative evaluations / Hessian-vector products.

The assumptions vary between results, but include the following:

- f and c are twice Lipschitz continuously differentiable.
- $f(x) + (\rho_0/2)\|c(x)\|_2^2$ has compact level sets, for some $\rho_0 \geq 0$.
- Constraint Jacobian $\nabla c(x)$ has uniformly full rank for all x. (Can be weakened to hold only on some compact level set.)

PAL Results I: Outer Iteration Complexity

For any $\epsilon > 0$ and $\eta \in [0, 2]$, choose

- Prox parameter $\beta = \epsilon^\eta$ (small);
- Penalty parameter $\rho = O(\epsilon^{-\eta})$ (large).

If we find an exact stationary point for each subproblem, Outer Iteration complexity to find an ϵ-1o point is $O(1/\epsilon^{2-\eta})$.

- When $\eta = 2$, need only $O(1)$ outer iterations! (But then the subproblem is extremely ill conditioned.)
- When $\eta = 0$ (settings of β and ρ are independent of ϵ), get $O(1/\epsilon^2)$ outer iterations.

Same complexity achieved for ϵ-2o point, for $\eta \in [1, 2]$, provided we find an exact second-order solution to each subproblem.

For inexact subproblem solution (next slides), Outer Iteration complexities are the same as above!

PAL Results II: Inexact, Total Iteration Complexity

Suppose we use line-search Newton-CG to solve the subproblems. Can then get estimates of total iteration complexity and operation complexity.

Subproblems solved inexactly, with square summable error sequence:

$$\nabla_x \mathcal{L}_\rho(x_{k+1}, \lambda_k) + \beta(x_{k+1} - x_k) = \tilde{r}_{k+1},$$
$$\nabla^2_{xx} \mathcal{L}_\rho(x_{k+1}, \lambda_k) + \beta I \succeq -\epsilon^H_{k+1} I,$$

where $\|\tilde{r}_{k+1}\| \leq \epsilon/2$ and $\sum_{k=1}^{+\infty} \|\tilde{r}_k\|^2 < +\infty$.

For $\epsilon^H_k \equiv \sqrt{\epsilon}/2$, $\eta \in [1, 2]$, and an ϵ-1o point:

Constraints	Total Iter. complexity	Optimal	
nonlinear	$\mathcal{O}(\epsilon^{-2\eta-7/2})$	$\mathcal{O}(\epsilon^{-11/2})$	$(\eta = 1)$
linear	$\mathcal{O}(\epsilon^{\eta-7/2})$	$\mathcal{O}(\epsilon^{-3/2})$	$(\eta = 2)$

For $\epsilon^H_k \equiv \epsilon/2$, $\eta \in [1, 2]$, ϵ-2o point w.h.p.:

Constraints	Total Iter. complexity	Optimal	
nonlinear	$\mathcal{O}(\epsilon^{-2\eta-5})$	$\mathcal{O}(\epsilon^{-7})$	$(\eta = 1)$
linear	$\mathcal{O}(\epsilon^{\eta-5})$	$\mathcal{O}(\epsilon^{-3})$	$(\eta = 2)$

PAL Results III: Inexact, Operation Complexity

For $\epsilon_k^H \equiv \sqrt{\epsilon}/2$, $\eta \in [1,2]$, and an ϵ-1o point:

Constraints	Total Iter. complexity	Optimal	
nonlinear	$\mathcal{O}(\epsilon^{-5\eta/2-15/4})$	$\mathcal{O}(\epsilon^{-25/4})$	$(\eta = 1)$
linear	$\mathcal{O}(\epsilon^{\eta/2-15/4})$	$\mathcal{O}(\epsilon^{-11/4})$	$(\eta = 2)$

For $\epsilon_k^H \equiv \epsilon/2$, $\eta \in [1,2]$, ϵ-2o point w.h.p.:

Constraints	Total Iter. complexity	Optimal	
nonlinear	$\mathcal{O}(\epsilon^{-5\eta/2-11/2})$	$\mathcal{O}(\epsilon^{-8})$	$(\eta = 1)$
linear	$\mathcal{O}(\epsilon^{\eta/2-11/2})$	$\mathcal{O}(\epsilon^{-9/2})$	$(\eta = 2)$

Summary

- Complexity analysis of nonconvex algorithms gives additional insight into their behavior and even suggests improvements that can help in some practical cases.
- Nonconvex problems are now ubiquitous in ML — but the performance of models algorithms is not well explained by classical theory.
- It's interesting to understand the nature of this benign nonconvexity is key applications.

THANKS!

References I


Agarwal, N., Allen-Zhu, Z., Bullins, B., Hazan, E., and Ma, T. (2017).
Finding approximate local minima faster than gradient descent.
arXiv:1611.01146v4.


Bhojanapalli, S., Neyshabur, B., and Srebro, N. (2016).
Global optimality of local search for low-rank matrix recovery.
Technical Report arXiv:1605.07221, Toyota Technological Institute.


Birgin, E. G. and Martínez, J. M. (2017).
The use of quadratic regularization with a cubic descent condition for unconstrained optimization.
SIAM J. Optim., 27:1049–1074.


Birgin, E. G. and Martínez, J. M. (2018).
On regularization and active-set methods with complexity for constrained optimization.
SIAM Journal on Optimization, 28(2):1367–1395.


Blanchet, J. and Murthy, K. (2019).
Quantifying distributional model risk via optimal transport.
Mathematics of Operations Research, 44(2):565–600.


Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. (2017a).
Accelerated methods for non-convex optimization.
arXiv:1611.00756v2.

References II

Carmon, Y., Duchi, J. C., Hinder, O., and Sidford, A. (2017b).
"Convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions.
In *Volume 70: International Conference on Machine Learning, 6-11 August 2017, International Convention Centre, Sydney, Australia*, pages 654–663. PMLR.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2011).
Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results.
Math. Program., 127:245–295.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2012).
An adaptive cubic regularization algorithm for nonconvex optimization with convex constraints and its function-evaluation complexity.
IMA Journal of Numerical Analysis, 32(4):1662–1695.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2015).
On the evaluation complexity of constrained nonlinear least-squares and general constrained nonlinear optimization using second-order methods.
SIAM Journal on Numerical Analysis, 53(2):836–851.

References III


Cartis, C., Gould, N. I. M., and Toint, P. L. (2018).
Sharp worst-case evaluation complexity bounds for arbitrary-order nonconvex optimization with inexpensive constraints.
arXiv preprint arXiv:1811.01220.


Chen, Z., Kuhn, D., and Wiesemann, W. (2018).
Data-Driven Chance Constrained Programs over Wasserstein Balls.
arXiv e-prints, page arXiv:1809.00210.


Chi, Y., Lu, Y. M., and Chen, Y. (2018).
Nonconvex optimization meets low-rank matrix factorization: An overview.
Technical Report arXiv:1809.09573, Carnegie Mellon University.


Curtis, F. E., Robinson, D. P., Royer, C. W., and Wright, S. J. (2019).
Trust-region Newton-CG with strong second-order complexity guarantees for nonconvex optimization.
Technical Report arXiv:1912.04365, Lehigh University.


Curtis, F. E., Robinson, D. P., and Samadi, M. (2017a).
An inexact regularized Newton framework with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization.
Technical Report 17T-011, COR@L Laboratory, Department of ISE, Lehigh University.

References IV

Curtis, F. E., Robinson, D. P., and Samadi, M. (2017b).

A trust region algorithm with a worst-case iteration complexity of $O\left(\epsilon^{-3/2}\right)$ for nonconvex optimization.

Math. Program., 162:1–32.

Fawzi, A., Fawzi, O., and Frossard, P. (2018).

Analysis of classifiers' robustness to adversarial perturbations.

Machine Learning, 107(3):481–508.

Gao, R. and Kleywegt, A. J. (2016).

Distributionally robust stochastic optimization with wasserstein distance.

Haeser, G., Liu, H., and Ye, Y. (2018).

Optimality condition and complexity analysis for linearly-constrained optimization without differentiability on the boundary.

Mathematical Programming.

Jin, C., Ge, R., Netrapalli, P., Kakade, S. M., and Jordan, M. I. (2017a).

How to escape saddle points efficiently.

arXiv:1703.00887v1.

Jin, C., Netrapalli, P., and Jordan, M. I. (2017b).

Accelerated gradient descent escapes saddle points faster than gradient descent.

arXiv:1711.10456.

References V


Kuczyński, J. and Woźniakowski, H. (1992).
Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start.
SIAM J. on Matrix Analysis and Applications, 13(4):1094–1122.


Kuczyński, J. and Woźniakowski, H. (1994).
Probabilistic bounds on the extremal eigenvalues and condition number by the Lanczos algorithm.
SIAM J. on Matrix Analysis and Applications, 15(2):672–691.


Martínez, J. M. and Raydan, M. (2017).
Cubic-regularization counterpart of a variable-norm trust-region method for unconstrained minimization.
J. Global Optim., 68:367–385.


Mohajerin Esfahani, P. and Kuhn, D. (2018).
Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations.
Mathematical Programming, 171(1):115–166.


Nesterov, Y. and Polyak, B. T. (2006).
Cubic regularization of Newton method and its global performance.
Mathematical Programming, Series A, 108:177–205.

References VI

O'Neill, M. and Wright, S. J. (2019).
A log-barrier Newton-CG method for bound constrained optimization with complexity guarantees.
Technical Report arXiv:1904.03563, University of Wisconsin-Madison.
Revised December 2019.

Royer, C. W., O'Neill, M., and Wright, S. J. (2018).
A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization.
Technical Report arXiv:1803.02924, University of Wisconsin-Madison.
To appear in *Mathematical Programming, Series A*.

Royer, C. W. and Wright, S. J. (2018).
Complexity analysis of second-order line-search algorithms for smooth nonconvex optimization.
SIAM Journal on Optimization, 28:1448–1477.

Xie, Y. and Wright, S. J. (2019).
Complexity of proximal augmented Lagrangian for nonconvex optimization with nonlinear equality constraints.
Technical Report arrXiv:1908.00131, University of Wisconsin-Madison.

Zhu, Z., Li, Q., Tang, G., and Wakin, M. B. (2017).
The global optimization geometry of low-rank matrix optimization.
Technical Report arXiv:1703.01256, Colorado School of Mines.

References VII