Optimization: Part II

Jorge Nocedal

Northwestern University

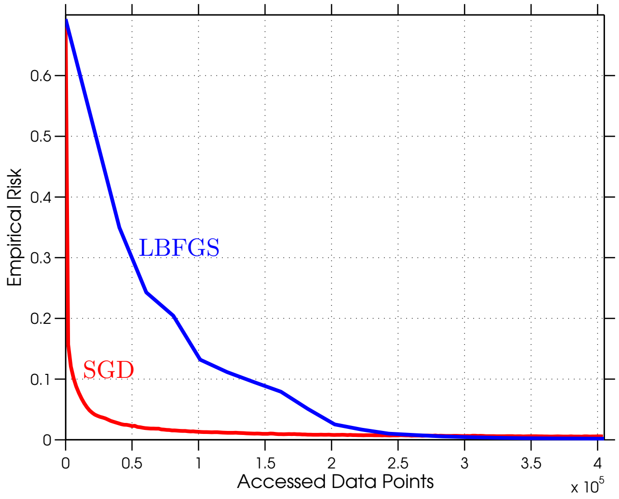
Toronto, July 2018

Many thanks to

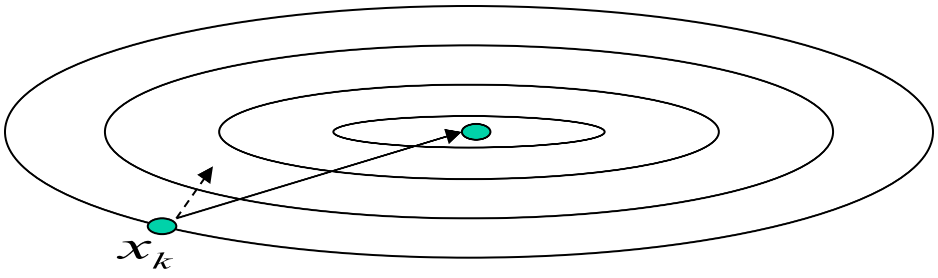
Albert Berahas
Raghu Bollapragada
Michael Shi

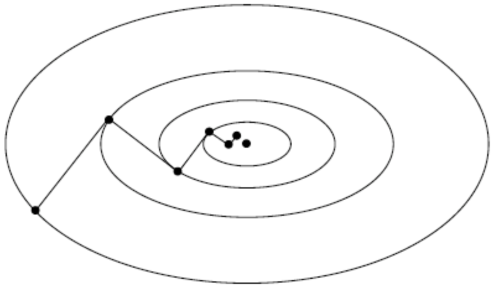
Different perspectives on optimization

- *"In the beginning there was SGD"*



- *"In the beginning there was Newton's method"*





Different perspectives on nonlinear optimization

- Russian school of optimization emphasized 1st order methods, convexity and complexity (Polyak, Nemirovski, Nesterov,…)
  - Yet it led to interior point methods that are 2nd order methods (Khachiyan)
- Western school focused early on on the use of second derivative information (Davidon 1959), convergence rates, non-convexity, and open source software (Fletcher –Powell)

- The above is an over-simplification (Rockafellar, Karmakar, many …) but it has some relevance today
- Both schools considered stochastic optimization problems (Robbins, Polyak)
- Deterministic vs Stochastic Optimization

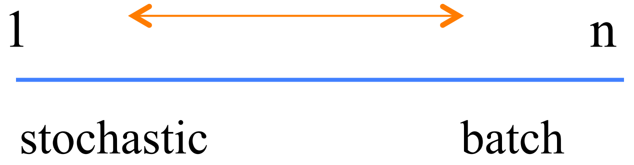
Deterministic and Stochastic Optimization

a) Large-scale nonlinear deterministic optimization, well researched:
   - Optimal trajectory, optimal design, etc

b) Stochastic optimization involves random variables (choice of data)
   - has borrowed ideas from the deterministic setting (gradient descent, momentum, preconditioning, etc.)

c) Exchange of ideas is not straightforward: stochastic approximation methods are different (Markov process).

d) There is a continuum: as quality of stochastic gradient approximation improves, algorithms resemble their deterministic counterparts



1 n

stochastic batch

e) The interplay between these two worlds – stochastic & deterministic – is ongoing.

f) This will be one of the unifying themes of this lecture

g) New algorithm originated from the ML community (Adagrad, ADAM) some inspiration from the deterministic setting

Stochastic and Deterministic Large-Scale Nonlinear Optimization Worlds

Stochastic
(Machine Learning)

SGD
SGD + momentum
ADAM
Adagrad
Natural gradient

Deterministic

Quasi-Newton
Inexact Newton
Nonlinear conjugate gradients

$$w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

$$w_{k+1} = w_k - \alpha_k H_k \nabla F_{X_k}(w_k)$$

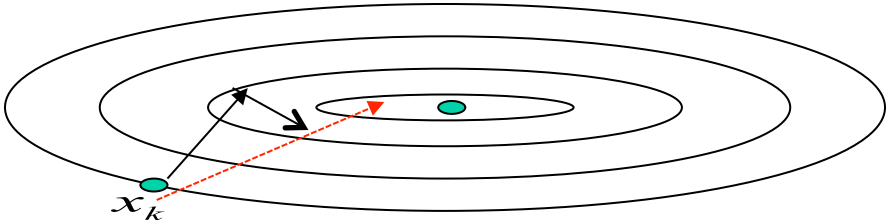
- first order methods
- empirical steplength rules
- inexpensive noisy iterations
- Fisher Information Matrix
    - Martens-Grosse (2016)

- simple gradient descent: not used
- acceleration & momentum: not used
- employ some 2nd order information using gradient differences
- line searches
- Hessian-vector products
- Hessian or Gauss-Newton matrices

To make this concrete let’s talk about
Momentum and Acceleration

Momentum (Heavy Ball Method)

$$w_{k+1} = w_k - \alpha_k \nabla F(w_k) + \beta_k (w_k - w_{k-1})$$



Beware of 2-d pictures!

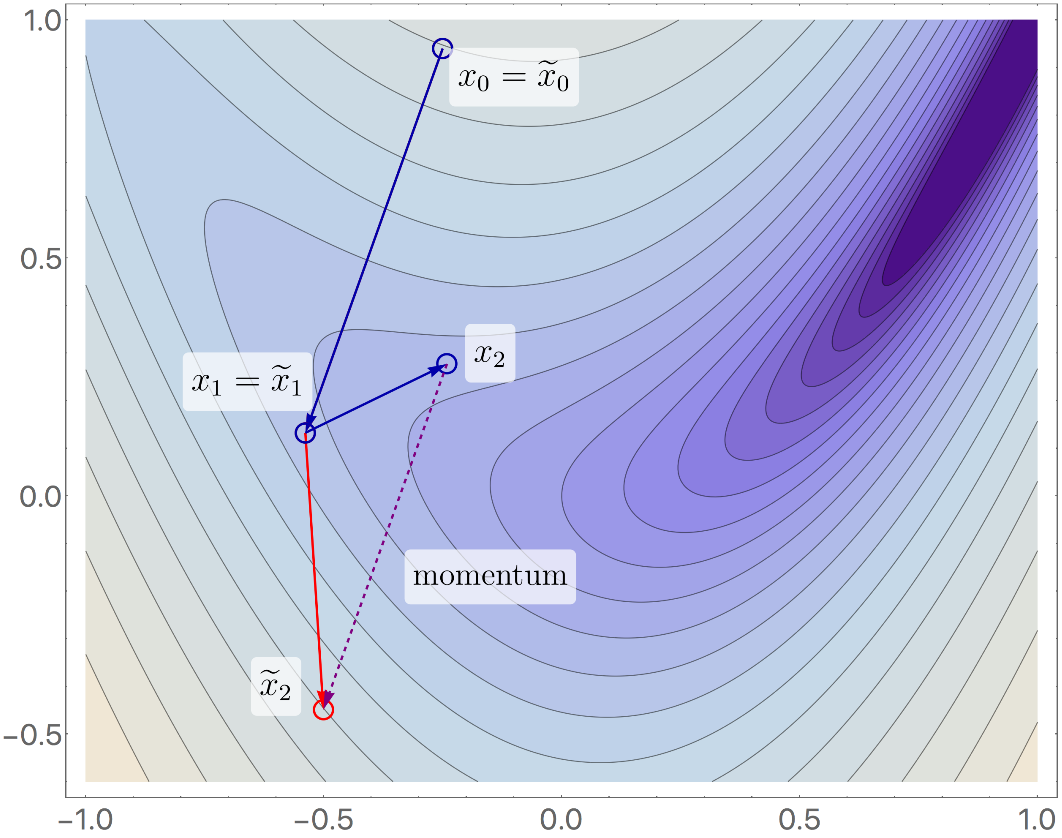
It is true that for convex quadratics the gradient method with momentum has a faster convergence rate than the pure gradient method
But:

- One needs a good estimate of the condition number of the Hessian

$$\alpha = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2} \quad \beta = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}$$

- DNN are not quadratics!
- Gradient method + momentum is not convergent on convex functions
- There are better iterative methods (CG) for quadratics

Consider what momentum can do in the non-convex case



Gradient method with momentum; $\beta = 0.9$

But momentum works in practice $w_{k+1} = w_k - \alpha_k \nabla F(w_k) + \beta_k (w_k - w_{k-1})$

- Popular since (Sutskever et al. 2013)
- Conjecture: it is not a real momentum method; neither a linear dynamical system with friction, nor Nesterov's optimal iteration
- Instead: a form of iterate (or gradient) averaging

$$\hat{w}_k = \sigma \hat{w}_{k-1} + (1-\sigma) w_k$$

- Gap between practice and algorithmic understanding
- Useful to compare with the Conjugate Gradient method

$$w_{k+1} = w_k + \alpha_k p_k \qquad p_k = -\nabla F(w_k) + \beta_k p_{k-1}$$

Designed for quadratic objective functions; easy to compute parameters
Same form as momentum but requires no estimate of condition number
For deterministic quadratic problems momentum is not better than CG
A version for nonlinear problems is available (PR+; see my website)

Nesterov acceleration

$$x_{k+1} = y_k - \alpha_k \nabla F(y_k)$$
$$y_{k+1} = x_k + \beta_k (x_{k+1} - x_k)$$

Remarkable result:

- If eigenvalue information is available
- Rate of convergence is $O((1 - \frac{1}{\sqrt{\kappa}})^k)$

- But is it relevant to practice? FISTA
  - Even for convex problems, can it compete with quasi-Newton method?
  - Suppose estimate of condition number is not accurate
- Many complexity papers on acceleration:
  - Find a stationary point, escaping saddles, combine with other methods, etc.
  - Very pessimistic results
- Would anyone use a method such as:
  - Apply Nesterov acceleration until function reveals to be non-convex, then estimate a negative eigenvalue and use it to generate a direction; Carmon et al. 2017

Acceleration with noisy gradients

- CG breaks downs
- For noisy gradients, momentum provides no benefits even for linear regression Kidambi et al 2018
- Benefits (very real) of momentum-type acceleration need to be investigated
- Highlighted in Jimmy Ba’s presentation

Understanding SGD

Convergence

$$w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

- Why does it converge, and for what classes of functions?
- Do they include DNNs or only some?

For deterministic convex optimization: $\min F(w)$

$$F(w_{k+1}) - F(w_k) \leq -\alpha_k \|\nabla F(w_k)\|_2^2$$

For stochastic problem: $\min F(w) \equiv \mathbb{E}[f(w;\xi)]$

$$\boxed{\mathbb{E}[F(w_{k+1}) - F(w_k)] \leq -\alpha_k \|\nabla F(w_k)\|_2^2 + \alpha_k^2 \mathbb{E}\|\nabla f(w_k, \xi_k)\|^2}$$

$$\Downarrow$$

$$\nabla F_{X_k}(w_k)$$

Two algorithmic components:

- $\nabla F_x(w_k)$ is an unbiased estimator of $\nabla F(w_k)$ (or good angle…)
- Steplength $\alpha_k \to 0$ and rate is sublinear $O(1/k)$

  Constant steplength $\alpha_k = \alpha$. Linear convergence to a neighborhood

Fixed steplength

Diminishing steplength

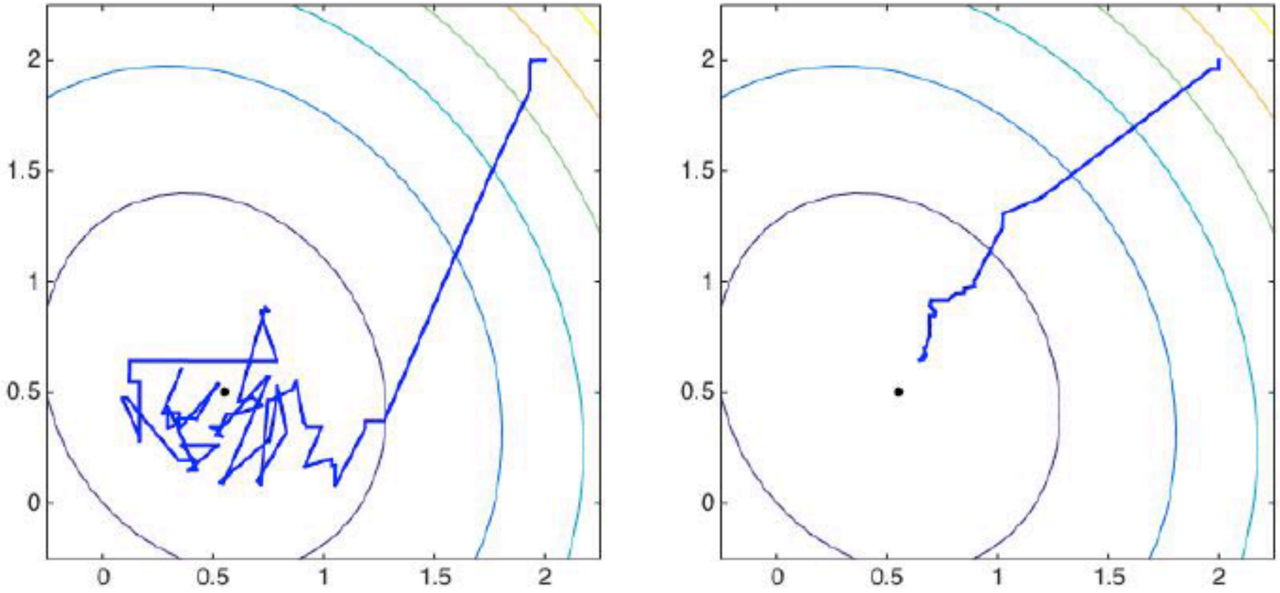


Figure: SG run with a fixed stepsize (left) vs. diminishing stepsizes (right)

Converges linearly to a neighborhood of the solution

Converges sub-linearly to the solution

Efficiency of SGD

$$w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

1. Why is SGD efficient on convex probems? Motivating examples (see SIAM Review paper by Bottou, Curtis, N (2018))
2. Jimmy Ba has outlined the main complexity results

Non-convexity and SGD

$$w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

1. Convergence: what is the meaning of the results? Are they useful? How do they compare with deterministic results for gradient method?
2. Complexity
3. Convergence to a saddle/minimizer (worst case bounds)
4. Escaping saddles (pessimistic)
5. Sanjeev Arora dsicussed these issues

Weaknesses of SGD?

$$w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

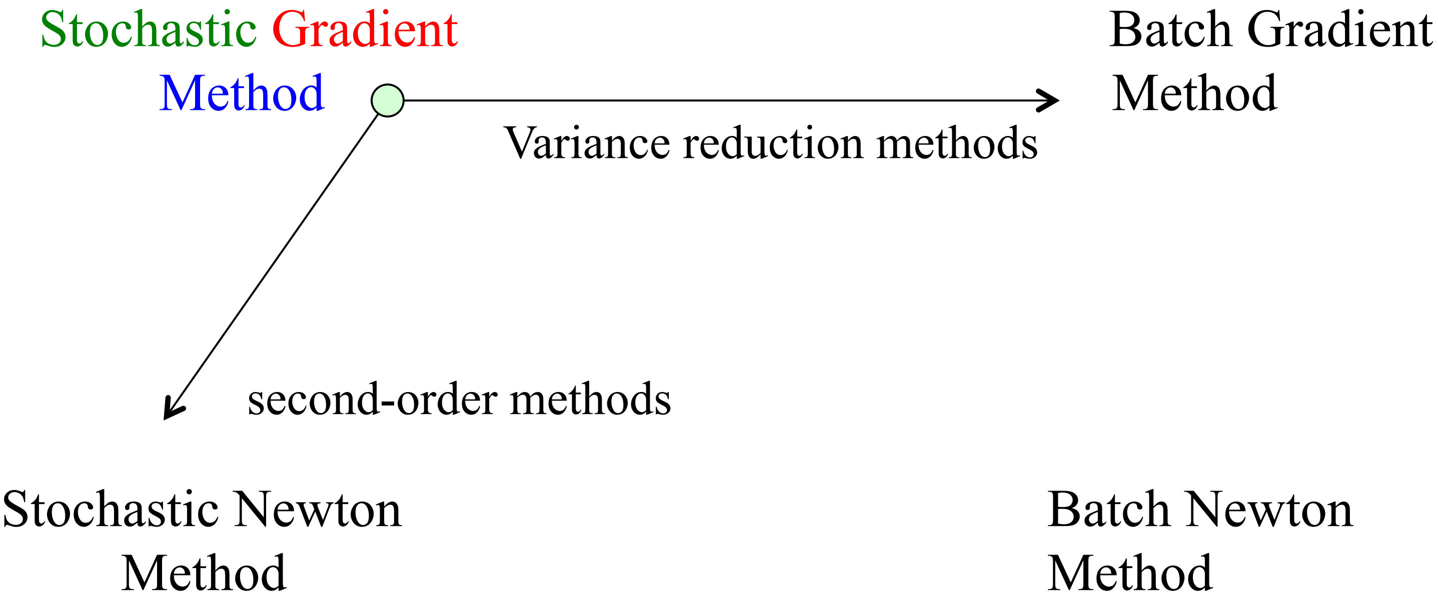
One view: Nothing. SGD (and variants) will not be improved

Alternate view: Various limitations

- Lack of scale; heuristic steplength selection, not solvable through universal formula – even with momentum terms
- Suffers from conditioning: it is a first order method
- Limited opportunities for parallelization

Optimization panorama

Bottou, Curtis, Nocedal 2018



Stochastic Gradient Method

Batch Gradient Method

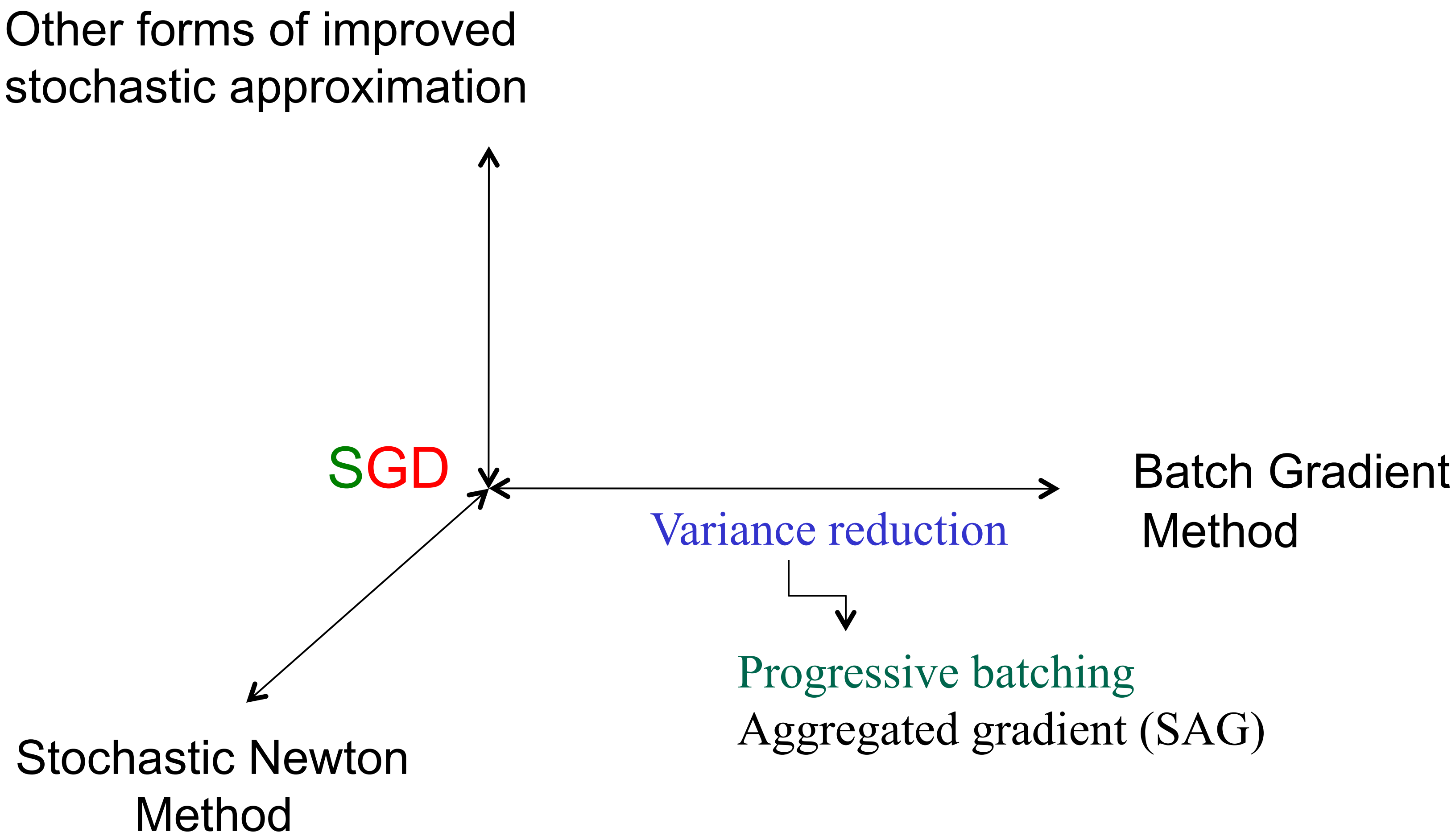
Variance reduction methods

second-order methods

Stochastic Newton Method

Batch Newton Method

$$\boxed{\mathbb{E}[F(w_{k+1}) - F(w_k)] \leq -\alpha_k \|\nabla F(w_k)\|_2^2 + \alpha_k^2 \, \mathbb{E}\|\nabla f(w_k, \xi_k)\|^2}$$



Other forms of improved stochastic approximation

SGD

Batch Gradient Method

Variance reduction

Progressive batching

Aggregated gradient (SAG)

Stochastic Newton Method

Three approaches for constructing second order information

- Inexact Newton Method with Hessian Sub-Sampling
- Natural Gradient Method (K-Fac)
- Quasi-Newton with Progressive Sampling

Mini-Batches

$$F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) \qquad \nabla F_{X_k}(w_k) = \frac{1}{|X_k|}\sum_{i\in X_k} \nabla f_i(w_k)$$

$X_k \subset \{1,2,\ldots\}$ drawn at random from distribution P.

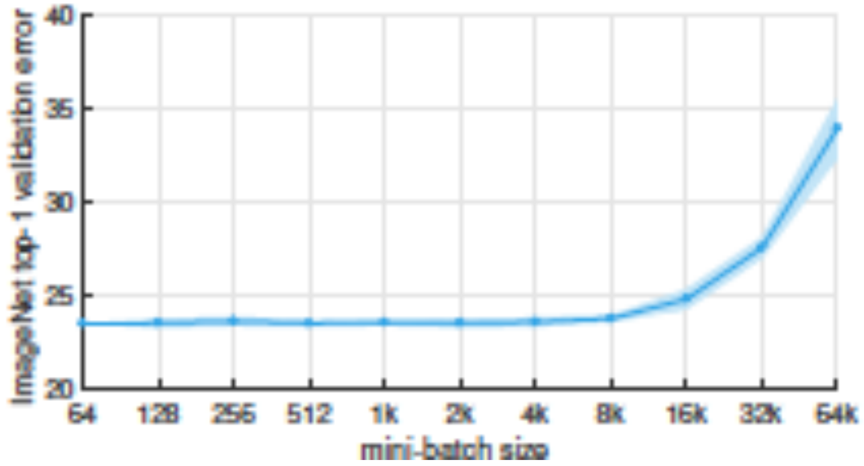
- Small (128) mini-batches standard; clearly useful
- Classical complexity theory does not show benefits of mini-batching (recently challenged)

- Why not use a gradient with a much larger batch, which enables data parallelism and the use of 2nd order information?
- Because as the batch size becomes larger, accuracy deteriorates (generalization) This has been observed for many years (empirically); a dozen recent systematic studies.
- Is SGD a regularizer?

The trade-offs of larger batch sizes

Accuracy

Residual Network, Imagenet



Batch size

N.B. Similar degradation occurs with stale updates in asynchronous SGD Chen et al. (2017)

Paper 1: Goyal et al. 2017: from 29 hours to 1 hour …
by increasing the batch size from 256 to 8k

Paper 2: Smith, Kindermans, Le (2017): batch of size 65k

- Instead of decreasing steplength, increase step size
- It is not well understood what causes the loss of accuracy
- A series of empirical observations

Robust minimizers (wide, sharp, entropy,…)

One conjecture:

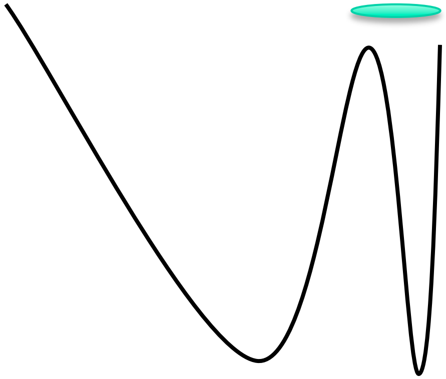
- SGD converges to robust minimizers in parameter & testing spaces
- SGD converges to points with large Jacobians in data space

Standard optimization problem: $\min_w F(w)$

Robust optimization problem:

$$\min_w \phi(w) \equiv \max_{\|\Delta x\| \le \epsilon} F(w + \Delta x)$$

much harder problem

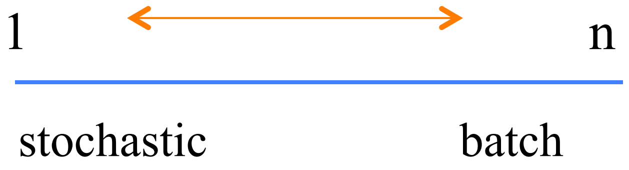


Progressive sampling gradient method

- Instead of manually choosing the mini-batch, or program a increase
- Develop an algorithmic procedure for gradually increasing the batch

$$\nabla F_{X_k}(w_k) = \frac{1}{|X_k|}\sum_{i\in X_k}\nabla f_i(w_k) \qquad\qquad w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

1. $|X_k| = 1$: stochastic gradient method
2. $|X_k| = n$: gradient method
3. $|X_k|$ grows as needed



- Noise in steps is controlled by sample size
- At the start, a small sample size $|X|$ is chosen
- If the optimization step is likely to reduce *F(w)*, sample size $|X|$ is kept unchanged; new sample *X* is chosen; next optimization step taken
- Else, a larger sample size is chosen, a new random sample *S* is selected, a new iterate computed

Progressive sampling gradient method

$$\nabla F_{X_k}(w_k) = \frac{1}{|X_k|}\sum_{i \in X_k} \nabla f_i(w_k) \qquad w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

- Many optimization methods can be used and this approach creates the opportunity of employing second order methods
- Crucial ingredient: rate at which sample is allowed to grow.

- Progressive batching gradient method matches work complexity of the SGD method by growing sample size geometrically $|X_k| = a^k, \quad a > 1$

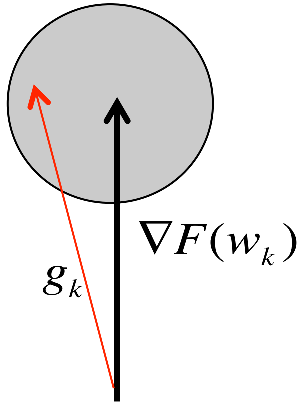
[Byrd, Chin, N., Wu 2013]

- Compare SGD with 1 sample vs progressive sampling method that increases X_k at a geometric rate
- Total work complexity to obtain an epsilon-accurate solution similar

How to use progressive sampling in practice?

Angle condition: $\nabla F(w_k)^T g_k > 0$

not most appropriate for probabilistic estimates



Proposed condition

$$\| g(w_k) - \nabla F(w_k) \| \le \theta \| g_k \| \qquad \theta < 1$$

which implies $\nabla F(w_k)^T g_k > 0$. Further:

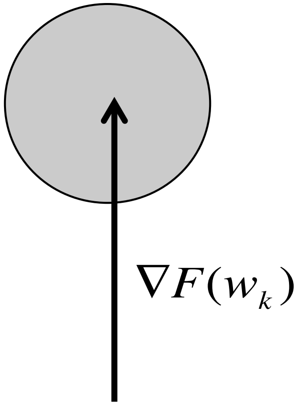
$$\frac{\| g(w_k) - \nabla F(w_k) \|}{\| g_k \|}$$

Is a quantity we can estimate if g(w) is an unbiased estimator used to create descent directions sufficiently often

Two strategies

Strategy I: Maintain batch size $|X_k|$ if

$$\frac{\mathbb{E}[\|\nabla F_i(w_k) - \nabla F(w_k)\|^2]}{|X_k|} \leq \theta^2 \|\nabla F(w_k)\|^2$$



Strategy II: only require

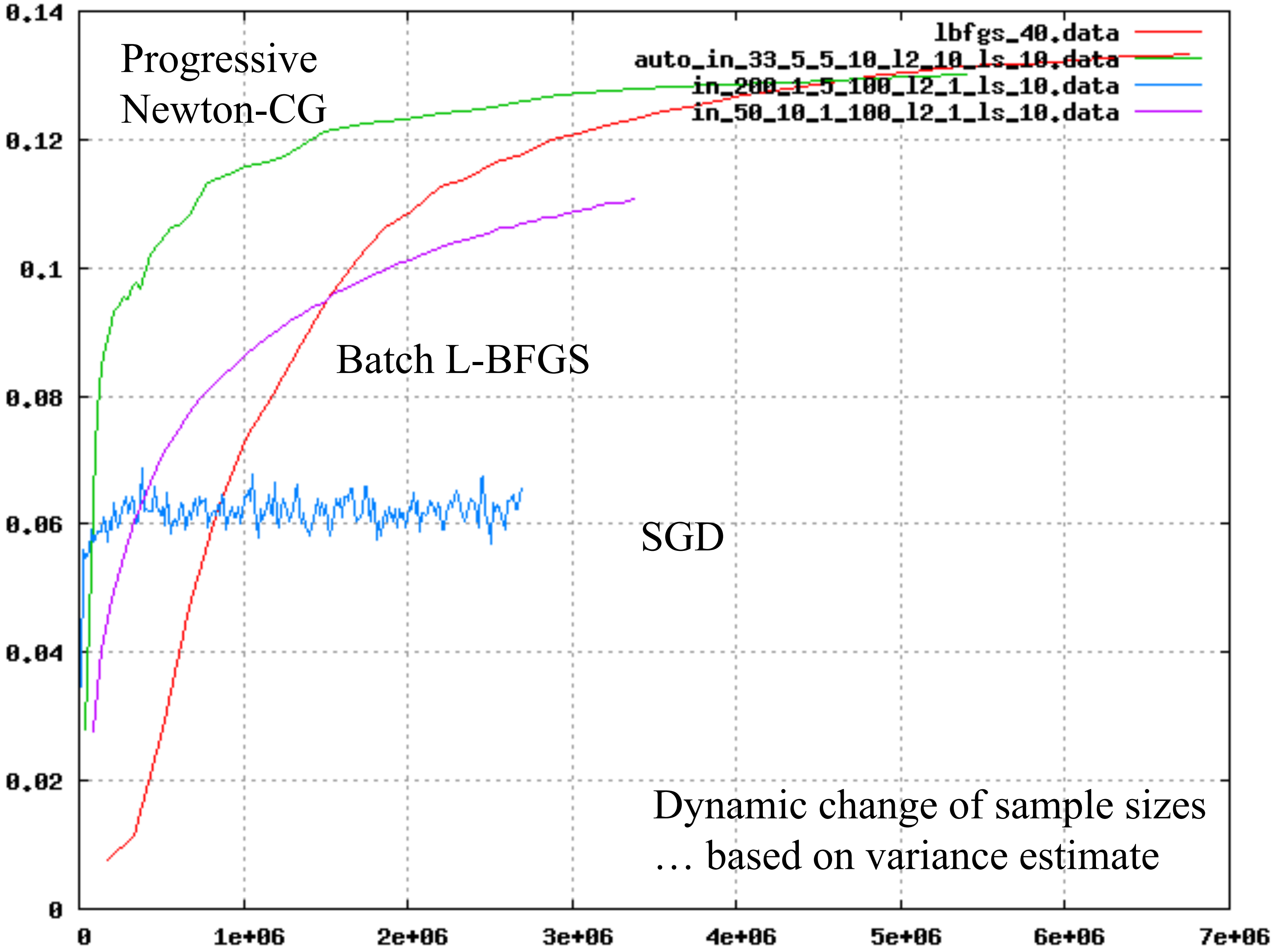
$$\frac{\mathbb{E}[(\nabla F_i(w_k)^T \nabla F(w_k) - \|\nabla F(w_k)\|^2)^2]}{|X_k|} \leq \theta^2 \|\nabla F(w_k)\|^4$$

- Gradient method with fixed steplength: Obtain linear (not sublinear) convergence to solution of convex problems

Implementation via sample variances

- Approximate population variance with sample variance and true gradient with sampled gradient

$$\frac{\mathrm{Var}_{i} \in X_k[(\nabla F_i(w_k)^T \nabla F(w_k))^2]}{|X_k|} \leq \theta^2 \|\nabla F(w_k)\|^4$$



Progressive
Newton-CG

Batch L-BFGS

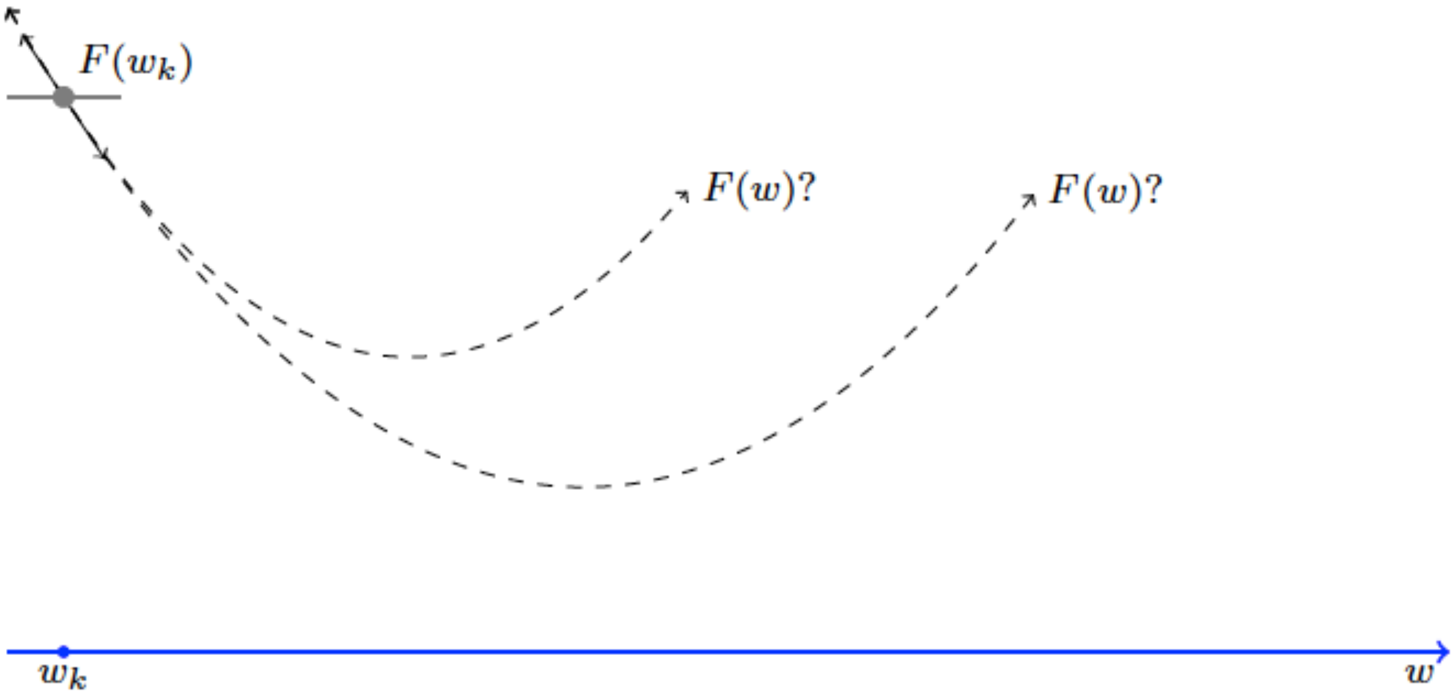
SGD

Dynamic change of sample sizes
… based on variance estimate

Time

On the Steplengths

$$w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

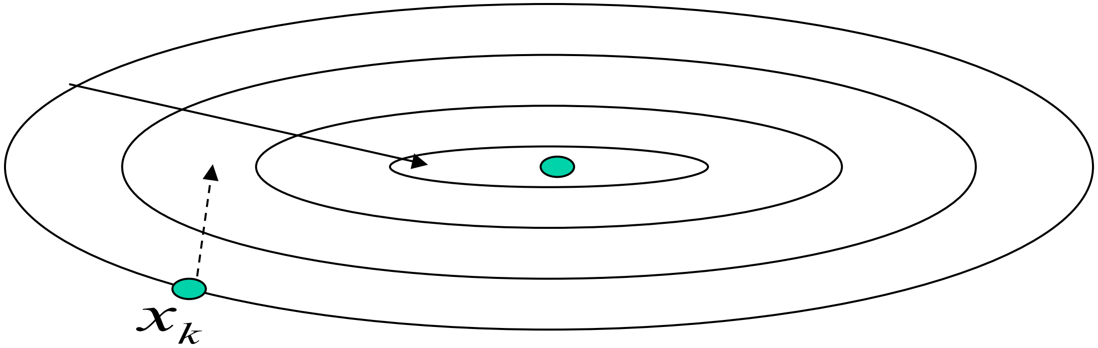


Scaling the Search Direction

- Different directions should be scaled differently
- For the noisy SGD method we will never find a formula for steplength that is universally practical
- Steplength tied up with noise suppression
- Mini-batching provides more freedom in choice of steplength

$$w_{k+1} = w_k - \alpha_k \nabla F_{X_k}(w_k)$$

Deterministic Setting

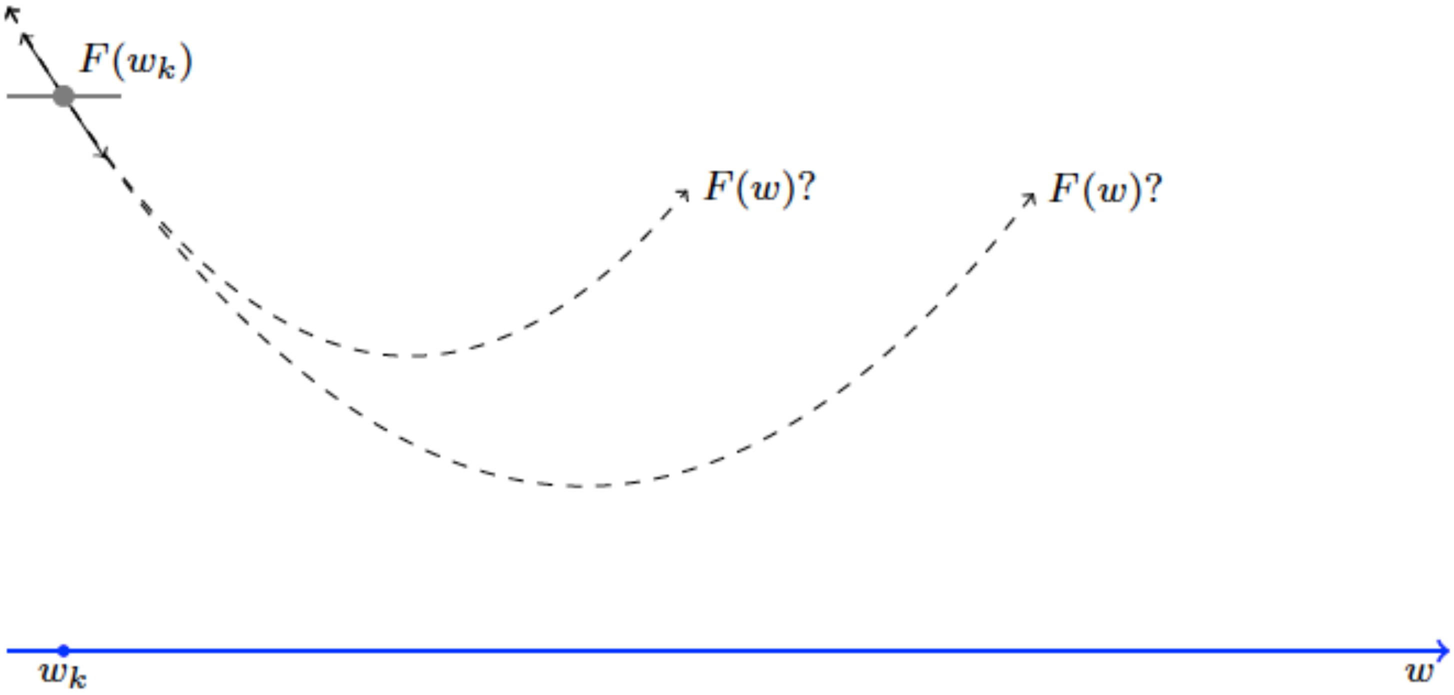


Scaling the Gradient Direction

Constant steplength (popular with theoreticians)

$$\alpha_k = 1/L \qquad L\text{: bound on } \|\nabla^2 F(w)\|$$

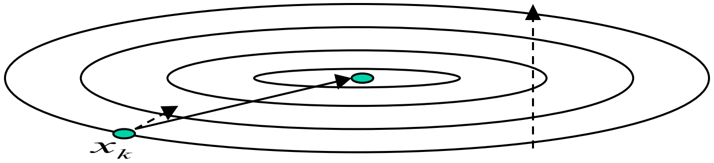
- Lipschitz constant *L*– the most conservative choice
- Adaptive (global) Lipschitz estimation – can be out of phase



Different gradient components should be scaled differently

$$w_{k+1} = w_k - \alpha_k D_k \nabla F_{X_k}(w_k)$$

1. Diagonal scaling (Adagrad, Adam)



2. Assumes knowledge along coordinate directions (difficult)
3. Generally not practical in deterministic optimization
4. Success of Adam and Adagrad explained through statistical arguments

Alternative:

- Instead of finding sophisticated steplength strategies, find method that produces well scaled directions
- Choice of steplength then becomes secondary
- Newton and quasi-Newton methods achieve this

Newton’s method

1. An ideal iteration: scale invariant, local quadratic rate of convergence

$$w_{k+1} = w_k - \alpha_k \nabla^2 F(w_k)^{-1} \nabla F(w_k)$$

2. The Hessian contains a lot of information, but too costly to form/invert
3. How to approximate Newton’s step?

Various Approaches:

1. Inexact Newton-CG – with subsampled Hessian
   - Computational unit: Hessian-vector product
2. Fischer Information – K-Fac Martens &Grosse 2017
3. Quasi-Newton – shows much potential
   - Computational unit: gradient
4. Tensor based block diagonal structures (Shampoo 2018)

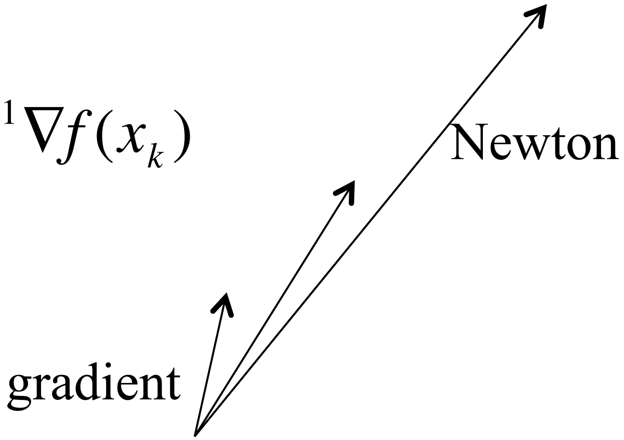
A Fundamental Equation for Newton's method

Strongly convex case (Hessian is positive definite)

$$\nabla^2 f(x_k) = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$ eigenvalue decomposition

$$\nabla^2 f(x_k)^{-1} = \sum_{i=1}^{n} \frac{1}{\lambda_i} v_i v_i^T \qquad p = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

$$\boxed{p = -\sum_{i=1}^{n} \frac{1}{\lambda_i} v_i (v_i^T \nabla f(x_k))}$$

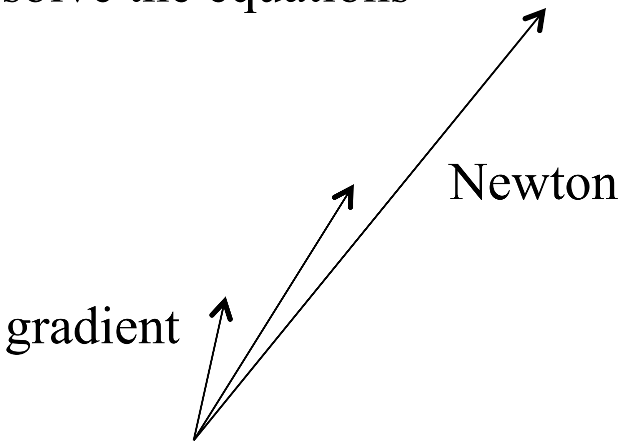


- direction points along eigenvectors corresponding to smallest eigenvalues
- Inexact Newton methods are based on this observation

Inexact Newton Method (Newton-CG)

… and on the important fact that an only matrix-vector products are needed by iterative methods like Conjugate Gradients to solve the equations

$$\nabla^2 f(x_k)p = -\nabla f(x_k)$$



A symmetric positive definite linear system
Many iterative methods; CG considered best
Increasing subspace minimization properties

Nonconvex Case:
Run CG until negative curvature is encountered; follow that direction
Sometimes called the Hessian-Free method in the ML community

Sub-sampled Hessian Newton Methods

Choose $X, S \subset \{1,2..,\}$, uniformly, independently from distribution P

$$\nabla F_X(w_k) = \frac{1}{|X|}\sum_{i\in X}\nabla f_i(w_k) \qquad \nabla^2 F_S(w_k) = \frac{1}{|S|}\sum_{i\in S}\nabla^2 f_i(w_k)$$

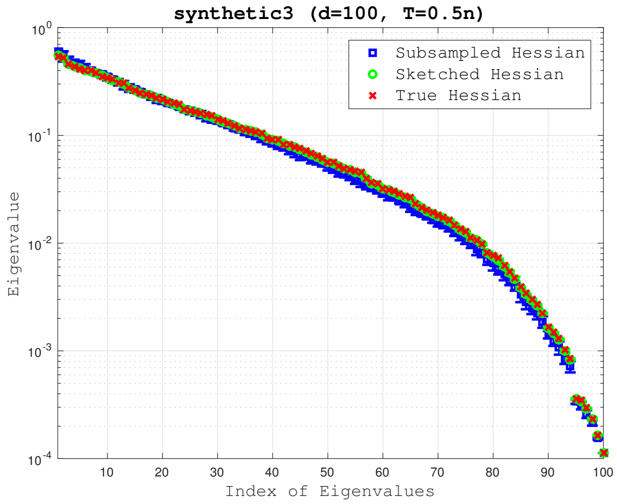
The stochastic nature of the objective creates opportunities:

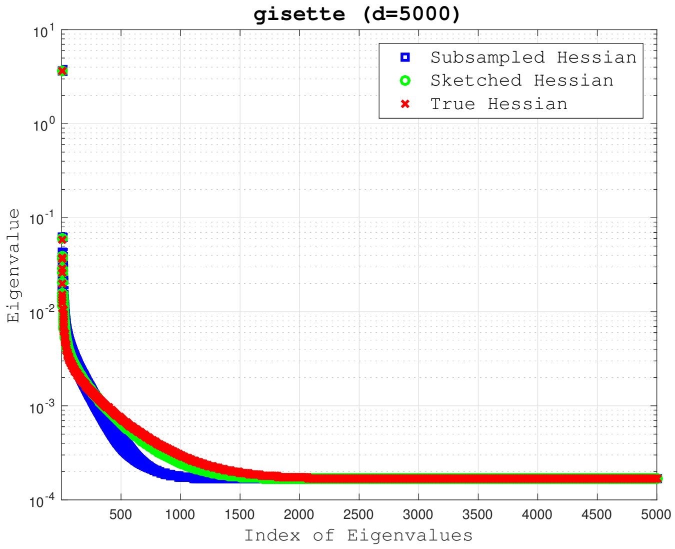
$$\boxed{\nabla^2 F_S(w_k)p = -\nabla F_X(w_k) \qquad w_{k+1} = w_k + \alpha_k p}$$

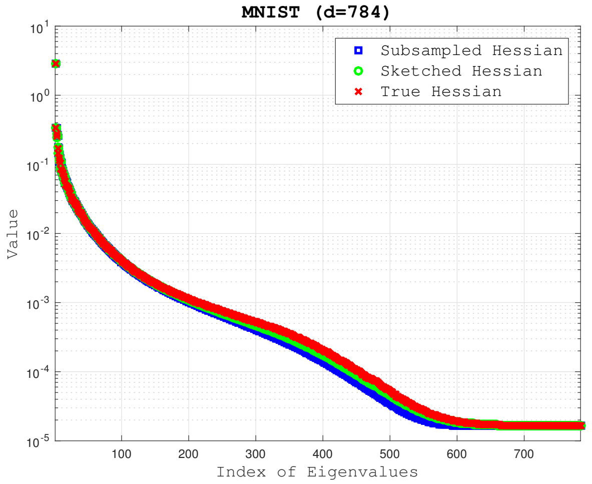
1. Subsampled gradient and Hessian (or other approximations)
2. How to coordinate choice of gradient and Hessian sampling?
3. Inexact solution of linear systems
4. What iterative method to use?
   - Conjugate gradient
   - Stochastic gradient   Bullins 2016, Neumann

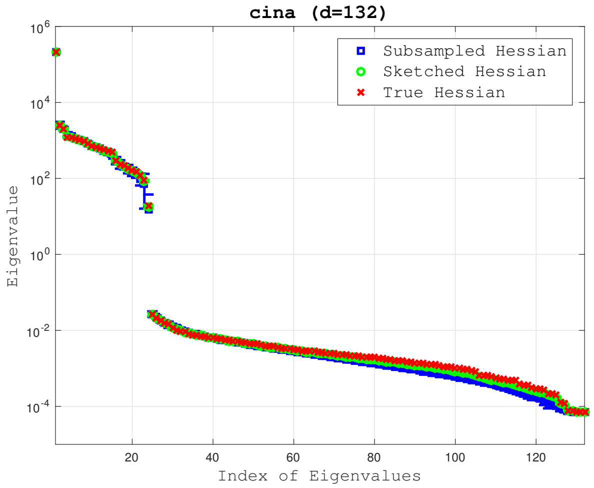
Eigenvalue Distribution of Hessian

Berahas, Bollapragada 2017









Active research area

- Martens (2010)
- Friedlander and Schmidt (2011)
- Byrd, Chin, Neveitt, N. (2011)
- Erdogdu and Montanari (2015)
- Roosta-Khorasani and Mahoney (2016)
- Byrd, Chin, N. Wu (2012)
- Agarwal, Bullins and Hazan (2016)
- Pilanci and Wainwright (2015)
- Pasupathy, Glynn, Ghosh, Hashemi (2015)
- Xu, Yang, Roosta-Khorasani, Re', Mahoney (2016)
- Cartis, Scheinberg (2016)
- Aravkin, Friedlander, Hermann, Van Leeuven (2012)

Local superlinear convergence

We can show the linear-quadratic result

$$\mathbb{E}_k[\| w_{k+1} - w^* \|] \leq \frac{M}{2\bar{\mu}} \| w_k - w^* \|^2 + \frac{\sigma \| w_k - w^* \|}{\mu\sqrt{|S_k|}} + \frac{v}{\mu\sqrt{|X_k|}}$$

To obtain superlinear convergence:

i) $|S_k| \to \infty$

ii) $|X_k|$ must increase faster than geometrically

In practice we are satisfied with linear convergence

Pilanci and Wainwright (2015)
Roosta-Khorasani and Mahoney (2016)
Bollapragada, Byrd, N (2016)

Inexact Methods- What is the best iterative solver?

$$\boxed{\nabla^2 F_S(w_k)p = -F_X(w_k) + b_k \qquad w_{k+1} = w_k + \alpha_k p}$$

1. Linear system solvers
   - Conjugate gradient
   - Stochastic gradient
2. Both require only Hessian-vector products

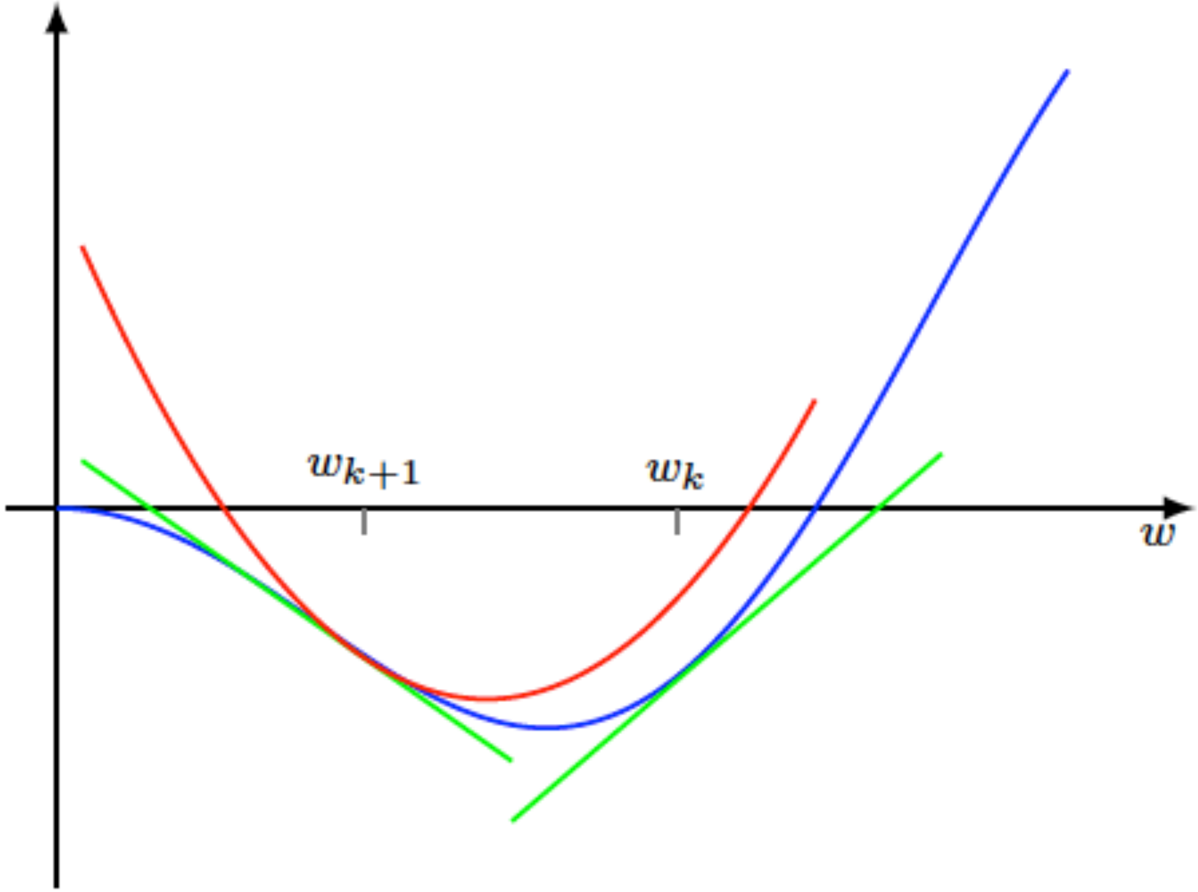
Quasi-Newton methods

A major idea in deterministic optimization

$$w_{k+1} = w_k - \alpha_k H_k \nabla F_{X_k}(w_k)$$

1. Learn curvature of problem on the fly through gradient differences
2. Incorporate curvature information that has been observed
3. Construct a dense Hessian approximation
4. Limited memory version L-BFGS avoids the use of matrices, requires storage and computation of *O(d)*

Only *approximate* second-order information with gradient displacements:



Secant equation $H_k v_k = s_k$ to match gradient of F at w_k, where

$$s_k := w_{k+1} - w_k \quad \text{and} \quad v_k := \nabla F(w_{k+1}) - \nabla F(w_k)$$

The BFGS method

Algorithm:

1. After performing a step, compute:

$$s \;= w_{k+1} - w_k \qquad y \;= \nabla F_x(w_{k+1}) - \nabla F_x(w_k)$$

2. $\rho \;= 1 / y^T s$

3. Update matrix:

$$\mathrm{H_k} = (I - \rho\ y\ s^T)H_{k-1}(I - \rho\ s\ y^T) + \rho\ s\ s^T$$

4. Search direction and iteration:

$$d_k = -H_k \nabla F_X(w_k) \qquad w_{k+1} = w_k + \alpha_k d_k$$

Progressive sampling quasi-Newton method Bollapragada, Shi 2018

$$w_{k+1} = w_k - \alpha_k H_k \nabla F_{X_k}(w_k)$$

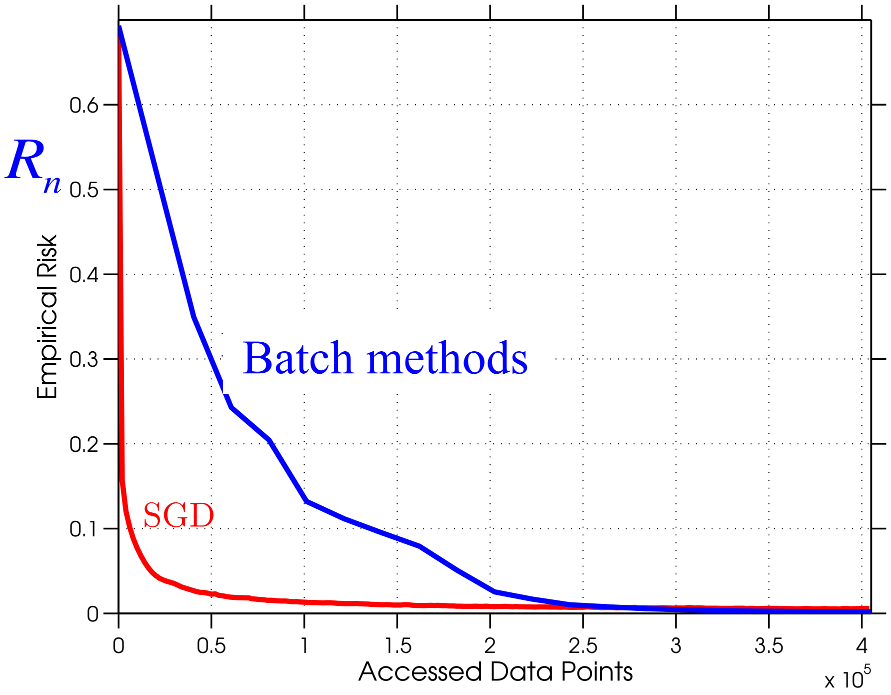
H_k updated by a careful (fault tolerant) form the the limited memory BFGS method

Line Search: Relaxing the sufficient decrease condition

$$F_{X_k}(w_k + \alpha_k p_k) \le F_{X_k}(w_k) + c_1 \alpha_k \nabla F_{X_k}(w_k)^T p_k + \epsilon_k$$

where ϵ_k is the noise level in the function

For years we observed this

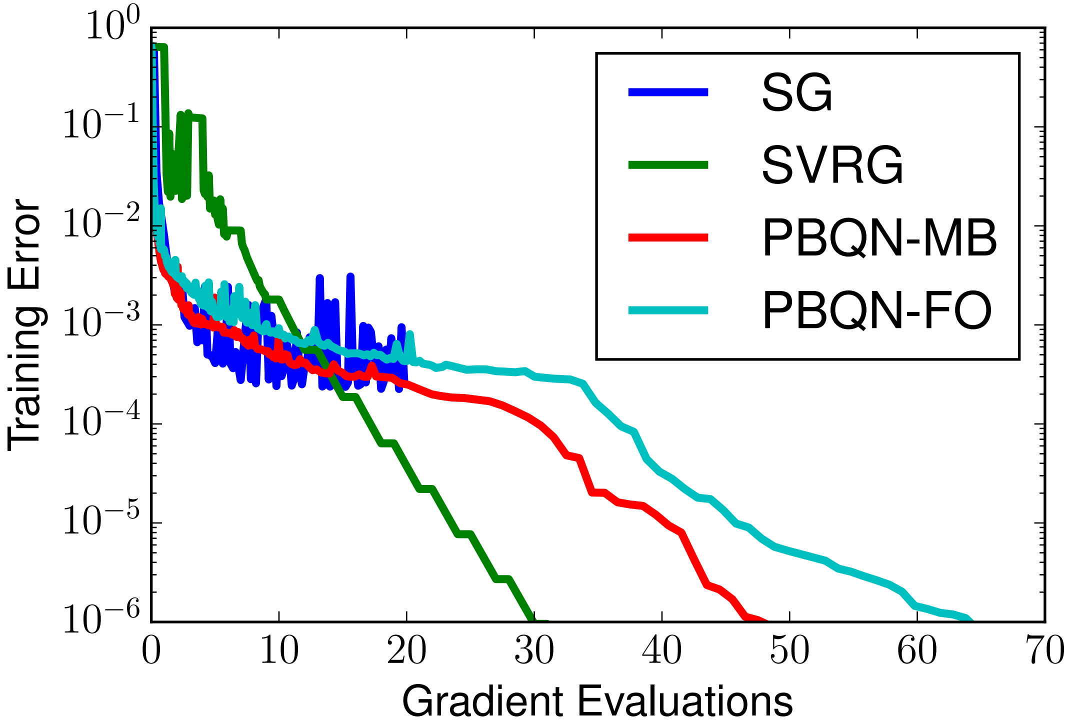


10 epochs

Logistic regression;
speech data

Fast initial progress
of SG followed by drastic
slowdown

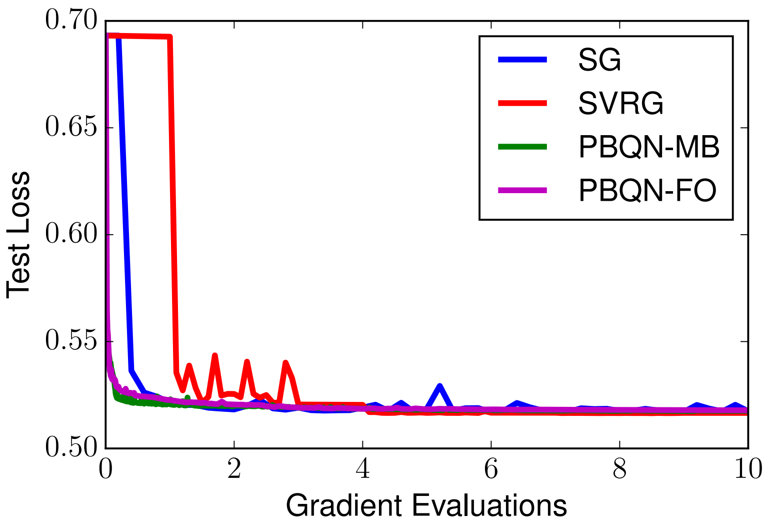
New Results: Progressive Sampling Quasi-Newton



Logistic Regression

- Results for DNN, in progress

Tests: Logistic Regression- Test Error



essive batching

-Newton method

- Stochastic quasi-Newton methods with noisy gradients in the typical regime of the SG method have not proved effective.
- Bollapragada, Shi et al (2018) have shown that a surprisingly small batch (100, 200) offers opportunities for quasi-Newton methods

End