CPSC 340: Machine Learning and Data Mining

Number of Iterations Gradient Descent

Mark Schmidt

University of British Columbia

Fall 2019

Cost of L2-Regularizd Least Squares

- Two strategies from 340 for L2-regularized least squares:
  1. Closed-form solution,
     $$w = (X^TX + \lambda I)^{-1}(X^Ty),$$
     which costs $O(nd^2 + d^3)$.
     - This is fine for $d = 5000$, but may be too slow for $d = 1,000,000$.
  2. Run t iterations of gradient descent,
     $$w^{k+1} = w^k - \alpha_k \underbrace{(X^T(Xw^k - y) + \lambda w^k)}_{\nabla f(w^k)},$$
     which costs $O(ndt)$.
     - I'm using t as total number of iterations, and k as iteration number.

- Gradient descent is faster if t is not too big:
  - If we only do $t < \max\{d, d^2/n\}$ iterations.

- So, how many iterations t of gradient descent do we need?

Outline

Gradient Descent for Finding a Local Minimum

- A typical gradient descent algorithm:
  - Start with some initial guess, w^0.
  - Generate new guess w^1 by moving in the negative gradient direction:

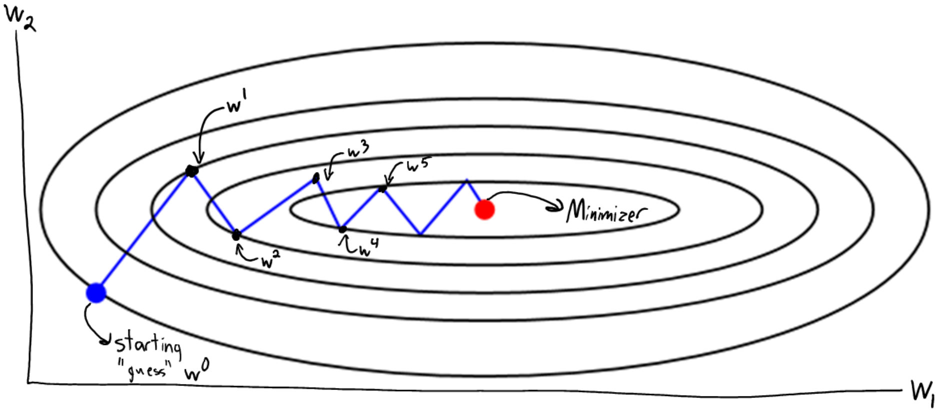
    $$w^1 = w^0 - \alpha_0 \nabla f(w^0),$$

    where α_0 is the step size.
  - Repeat to successively refine the guess:

    $$w^{k+1} = w^k - \alpha_k \nabla f(w^k), \quad \text{for } k = 1, 2, 3, \dots$$

    where we might use a different step-size α_k on each iteration.
  - Stop if $\|\nabla f(w^k)\| \leq \epsilon$.
    - In practice, you also stop if you detect that you aren't making progress.

Gradient Descent in 2D



Lipschitz Contuity of the Gradient

- Let's first show a basic property:
  - If the step-size α_t is small enough, then gradient descent decreases f.

- We'll analyze gradient descent assuming gradient of f is Lipschitz continuous.
  - There exists an L such that for *all* w and v we have

$$\|\nabla f(w) - \nabla f(v)\| \leq L\|w - v\|.$$

  - "Gradient can't change arbitrarily fast".

- This is a fairly weak assumption: it's true in almost all ML models.
  - Least squares, logistic regression, neural networks with sigmoid activations, etc.

Lipschitz Contuity of the Gradient

- For C^2 functions, Lipschitz continuity of the gradient is equivalent to

$$\nabla^2 f(w) \preceq LI,$$

  for all w.
- Equivalently: "singular values of the Hessian are bounded above by L".
  - For least squares, minimum L is the maximum eigenvalue of $X^T X$.
- This means we can bound quadratic forms involving the Hessian using

$$\begin{aligned} d^T \nabla^2 f(u) d &\leq d^T (LI) d \\ &= L d^T d \\ &= L\|d\|^2. \end{aligned}$$

Descent Lemma

- For a C^2 function, a variation on the multivariate Taylor expansion is that

$$f(v) = \underbrace{f(w) + \nabla f(w)^T(v-w)}_{\text{tangent hyper-plane}} + \underbrace{\frac{1}{2}(v-w)^T\nabla^2 f(u)(v-w)}_{\text{quadratic form}},$$

  for any w and v (with u being some convex combination of w and v).

- Lipschitz continuity implies the green term is at most $L\|v-w\|^2$,

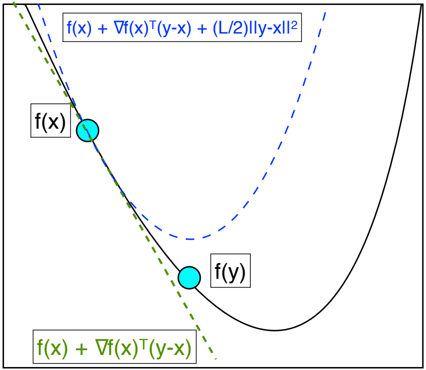
$$f(v) \leq f(w) + \nabla f(w)^T(v-w) + \frac{L}{2}\|v-w\|^2,$$

  which is called the descent lemma.

- The descent lemma also holds for C^1 functions (bonus slide).

Descent Lemma

- The descent lemma gives us a convex quadratic upper bound on f:



- This bound is minimized by a gradient descent step from w with $\alpha_k = 1/L$.

Gradient Descent decreases f for $\alpha_k = 1/L$

- So let's consider doing gradient descent with a step-size of $\alpha_k = 1/L$,

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

- If we substitle w^{k+1} and w^k into the descent lemma we get

$$f(w^{k+1}) \leq f(w^k) + \nabla f(w^k)^T(w^{k+1} - w^k) + \frac{L}{2}\|w^{k+1} - w^k\|^2.$$

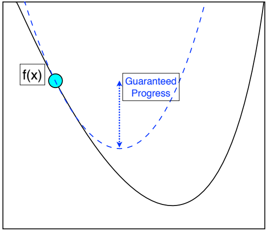
- Now if we use that $(w^{k+1} - w^k) = -\frac{1}{L}\nabla f(w^k)$ in gradient descent,

$$\begin{aligned} f(w^{k+1}) &\leq f(w^k) - \frac{1}{L}\nabla f(w^k)^T\nabla f(w^k) + \frac{L}{2}\|\frac{1}{L}\nabla f(w^k)\|^2 \\ &= f(w^k) - \frac{1}{L}\|\nabla f(w^k)\|^2 + \frac{1}{2L}\|\nabla f(w^k)\|^2 \\ &= f(w^k) - \frac{1}{2L}\|\nabla f(w^k)\|^2. \end{aligned}$$

Implication of Lipschitz Continuity

- We've derived a bound on guaranteed progress when using $\alpha_k = 1/L$.

$$f(w^{k+1}) \leq f(w^k) - \frac{1}{2L}\|\nabla f(w^k)\|^2.$$



- If gradient is non-zero, $\alpha_k = 1/L$ is guaranteed to decrease objective.
- Amount we decrease grows with the size of the gradient.
- Same argument shows that any $\alpha_k < 2/L$ will decrease f.

Choosing the Step-Size in Practice

- In practice, you should never use $\alpha_k = 1/L$.
  - L is usually expensive to compute, and this step-size is really small.
    - You only need a step-size this small in the worst case.
- One practical option is to approximate L:
  - Start with a small guess for $\hat{L}$ (like $\hat{L} = 1$).
  - Before you take your step, check if the progress bound is satisfied:

$$f(\underbrace{w^k - (1/\hat{L})\nabla f(w^k)}_{\text{potential } w^{k+1}}) \leq f(w^k) - \frac{1}{2\hat{L}}\|\nabla f(w^k)\|^2.$$

  - Double $\hat{L}$ if it's not satisfied, and test the inequality again.
- Worst case: eventually have $L \leq \hat{L} < 2L$ and you decrease f at every iteration.
- Good case: $\hat{L} << L$ and you are making more progress than using $1/L$.

Choosing the Step-Size in Practice

- An approach that usually works better is a backtracking line-search:
  - Start each iteration with a large step-size α.
    - So even if we took small steps in the past, be optimistic that we're not in worst case.
  - Decrease α until if Armijo condition is satisfied (this is what *findMin.jl* does),

$$f(\underbrace{w^k - \alpha\nabla f(w^k)}_{\text{potential } w^{k+1}}) \leq f(w^k) - \alpha\gamma\|\nabla f(w^k)\|^2 \quad \text{for} \quad \gamma \in (0, 1/2],$$

  often we choose γ to be very small like $\gamma = 10^{-4}$.
    - We would rather take a small decrease instead of trying many α values.

- Good codes use clever tricks to initialize and decrease the α values.
  - Usually only try 1 value per iteration.
- Even more fancy line-search: Wolfe conditions (makes sure α is not too small).
  - Good reference on these tricks: Nocedal and Wright's Numerical Optimization book.

Outline

Convergence Rate of Gradient Descent

- In 340, we claimed that $\nabla f(w^k)$ converges to zero as k goes to ∞.
  - For convex functions, this means it converges to a global optimum.
  - However, we may not have $\nabla f(w^k) = 0$ for any finite k.

- Instead, we're usually happy with $\|\nabla f(w^k)\| \leq \epsilon$ for some small ϵ.
  - Given an ϵ, how many iterations does it take for this to happen?

- We'll first answer this question only assuming that
  1. Gradient ∇f is Lipschitz continuous (as before).
  2. Step-size $\alpha_k = 1/L$ (this is only to make things simpler).
  3. Function f can't go below a certain value f^* ("bounded below").

- Most ML objectives f are bounded below (like the squared error being at least 0).
  - We're not assuming convexity (argument will work for any smooth problem).

Convergence Rate of Gradient Descent

- Key ideas:
  1. We start at some $f(w^0)$, and at each step we decrease f by at least $\frac{1}{2L}\|\nabla f(w^k)\|^2$.
  2. But we can't decrease $f(w^k)$ below f^*.
  3. So $\|\nabla f(w^k)\|^2$ must be going to zero "fast enough".

- Let's start with our guaranteed progress bound,

$$f(w^k) \leq f(w^{k-1}) - \frac{1}{2L}\|\nabla f(w^{k-1})\|^2.$$

- Since we want to bound $\|\nabla f(w^k)\|$, let's rearrange as

$$\|\nabla f(w^{k-1})\|^2 \leq 2L(f(w^{k-1}) - f(w^k)).$$

Convergence Rate of Gradient Descent

- So for each iteration k, we have

$$\|\nabla f(w^{k-1})\|^2 \leq 2L[f(w^{k-1}) - f(w^k)].$$

- Let's sum up the squared norms of all the gradients up to iteration t,

$$\sum_{k=1}^{t} \|\nabla f(w^{k-1})\|^2 \leq 2L \sum_{k=1}^{t}[f(w^{k-1}) - f(w^k)].$$

- Now we use two tricks:
  1. On the left, use that all $\|\nabla f(w^{k-1})\|$ are at least as big as their minimum.
  2. On the right, use that this is a telescoping sum:

$$\sum_{k=1}^{t}[f(w^{k-1}) - f(w^k)] = f(w^0) - \underbrace{f(w^1) + f(w^1)}_{0} - \underbrace{f(w^2) + f(w^2)}_{0} - \dots f(w^t)$$

$$= f(w^0) - f(w^t).$$

Convergence Rate of Gradient Descent

- With these substitutions we have

$$\sum_{k=1}^{t} \underbrace{\min_{j \in \{0,\dots,t-1\}} \{\|\nabla f(w^j)\|^2\}}_{\text{no dependence on } k} \leq 2L[f(w^0) - f(w^t)].$$

- Now using that $f(w^t) \geq f^*$ we get

$$t \min_{k \in \{0,1,\dots,t-1\}} \left\{\|\nabla f(w^k)\|^2\right\} \leq 2L[f(w^0) - f^*],$$

and finally that

$$\min_{k \in \{0,1,\dots,t-1\}} \left\{\|\nabla f(w^k)\|^2\right\} \leq \frac{2L[f(w^0) - f^*]}{t} = O(1/t),$$

so if we run for t iterations, we'll find $\underbrace{\text{least one } k}_{\text{the minimum}}$ with $\|\nabla f(w^k)\|^2 = O(1/t)$.

Convergence Rate of Gradient Descent

- Our "error on iteration t" bound:

$$\min_{k\in\{0,1,\dots,t-1\}}\left\{\|\nabla f(w^k)\|^2\right\} \leq \frac{2L[f(w^0)-f^*]}{t}.$$

- We want to know when the norm is below ϵ, which is guaranteed if:

$$\frac{2L[f(w^0)-f^*]}{t} \leq \epsilon.$$

- Solving for t gives that this is guaranteed for every t where

$$t \geq \frac{2L[f(w^0)-f^*]}{\epsilon},$$

so gradient descent requires $t = O(1/\epsilon)$ iterations to achieve $\|\nabla f(w^k)\|^2 \leq \epsilon$.

Discussion of $O(1/t)$ and $O(1/\epsilon)$ Results

- So if computing gradient costs $O(nd)$, total cost of gradient descent is $O(nd/\epsilon)$.
  - $O(nd)$ per iteration and $O(1/\epsilon)$ iterations.
- This also be shown for practical step-size strategies.
  - Just changes constants.
- This convergence rate is dimension-independent:
  - It does not directly depend on dimension d.
  - Though L might grow as dimension increases.
- Consider least squares with a fixed L and $f(w^0)$, and an accuracy ϵ:
  - There is dimension d beyond which gradient descent is faster than normal equations.

Outline

Iteration Complexity

- Iteration complexity: smallest t such that algorithm guarantees ϵ-solution.

- Think of $\log(1/\epsilon)$ as "number of digits of accuracy" you want.
    - We want iteration complexity to grow slowly with $1/\epsilon$.

- Is $O(1/\epsilon)$ a good iteration complexity?

- Not really, if you need 10 iterations for a "digit ''of accuracy then:
    - You might need 100 for 2 digits.
    - You might need 1000 for 3 digits.
    - You might need 10000 for 4 digits.

- We would normally call this exponential time.

Polyak-Łojasiewicz (PL) Inequality

- In scientific computing, having an error like $O(1/t)$ is called a sublinear rate.
- For many "nice" functions f, gradient descent actually has a linear rate.
  - Error is $O(\rho^t)$ after t iterations, so we only need $O(\log(1/\epsilon))$ iterations.
  - This is more like a polynomial number of iterations.
- For example, for functions satisfying the Polyak-Łojasiewicz (PL) inequality,

$$\frac{1}{2}\|\nabla f(w)\|^2 \geq \mu(f(w) - f^*),$$

for all w and some $\mu > 0$.
  - "Gradient grows as a quadratic function as we increase f".

Linear Convergence under the PL Inequality

- Recall our guaranteed progress bound

$$f(w^{k+1}) \leq f(w^k) - \frac{1}{2L}\|\nabla f(w^k)\|^2.$$

- Under the PL inequality we have $-\|\nabla f(w^k)\|^2 \leq -2\mu(f(w^k) - f^*)$, so

$$f(w^{k+1}) \leq f(w^k) - \frac{\mu}{L}(f(w^k) - f^*).$$

- Let's subtract f^* from both sides,

$$f(w^{k+1}) - f^* \leq f(w^k) - f^* - \frac{\mu}{L}(f(w^k) - f^*),$$

and factorizing the right side gives

$$f(w^{k+1}) - f^* \leq \left(1 - \frac{\mu}{L}\right)(f(w^k) - f^*).$$

Linear Convergence under the PL Inequality

- Applying this recursively:

$$
\begin{aligned}
f(w^k) - f^* &\leq \left(1 - \frac{\mu}{L}\right)[f(w^{k-1}) - f(w^*)] \\
&\leq \left(1 - \frac{\mu}{L}\right)\left[\left(1 - \frac{\mu}{L}\right)[f(w^{k-2}) - f^*]\right] \\
&= \left(1 - \frac{\mu}{L}\right)^2 [f(w^{k-2}) - f^*] \\
&\leq \left(1 - \frac{\mu}{L}\right)^3 [f(w^{k-3}) - f^*] \\
&\leq \left(1 - \frac{\mu}{L}\right)^k [f(w^0) - f^*]
\end{aligned}
$$

- We'll always have $0 < \mu \leq L$ so we have $(1 - \mu/L) < 1$.
  - So PL implies a linear convergence rate: $f(w^k) - f^* = O(\rho^k)$ for $\rho < 1$.

Linear Convergence under the PL Inequality

- We've shown that

$$f(w^k) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k [f(w^0) - f^*]$$

- By using the inequality that

$$(1 - \gamma) \leq \exp(-\gamma),$$

  we have that

$$f(w^k) - f^* \leq \exp\left(-k\frac{\mu}{L}\right)[f(w^0) - f^*],$$

  which is why linear convergence is sometimes called "exponential convergence".

- We'll have $f(w^t) - f^* \leq \epsilon$ for any t where

$$t \geq \frac{L}{\mu}\log((f(w^0) - f^*)/\epsilon) = O(\log(1/\epsilon)).$$

Discussion of Linear Convergence under the PL Inequality

- PL is satisfied for many standard convex models like least squares (bonus).
  - So cost of least squares is $O(nd\log(1/\epsilon))$.

- PL is also satisfied for some non-convex functions like $w^2 + 3\sin^2(w)$.
  - It's satisfied for PCA on a certain "Riemann manifold".
  - But it's not satisfied for many models, like neural networks.

- The PL constant μ might be terrible.
  - For least squares μ is the smallest non-zero eigenvalue of the Hessian.
- It may be hard to show that a function satisfies PL.
  - But regularizing a convex function gives a PL function with non-trivial μ...

Strong Convexity

- We say that a function f is strongly convex if the function

$$f(w) - \frac{\mu}{2}\|w\|^2,$$

  is a convex function for some $\mu > 0$.
  - "If you 'un-regularize' by μ then it's still convex."

- For C^2 functions this is equivalent to assuming that

$$\nabla^2 f(w) \succeq \mu I,$$

  that the eigenvalues of the Hessian are at least μ everywhere.

- Two nice properties of strongly-convex functions:
  - A unique solution exists.
  - C^1 strongly-convex functions satisfy the PL inequality with constant μ (bonus).

Effect of Regularization on Convergence Rate

- We said that f is strongly convex if the function

$$f(w) - \frac{\mu}{2}\|w\|^2,$$

  is a convex function for some $\mu > 0$.
    - For a C^2 univariate function, equivalent to $f''(w) \geq \mu$.

- If we have a convex loss f, adding L2-regularization makes it strongly-convex,

$$f(w) + \frac{\lambda}{2}\|w\|^2,$$

  with strong-convexity (and PL constant) μ being at least λ.

- So adding L2-regularization can improve rate from sublinear to linear.
    - Go from exponential $O(1/\epsilon)$ to polynomial $O(\log(1/\epsilon))$ iterations.
    - And guarantees a unique solution.

Effect of Regularization on Convergence Rate

- Our convergence rate under PL was

$$f(w^k) - f^* \leq \underbrace{\left(1 - \frac{\mu}{L}\right)^k}_{\rho^k} [f(w^0) - f^*].$$

- For L2-regularized least squares we have

$$\frac{L}{\mu} = \frac{\max\{\text{eig}(X^\top X)\} + \lambda}{\min\{\text{eig}(X^\top X)\} + \lambda}.$$

- So as λ gets larger ρ gets closer to 0 and we converge faster.

- The number $\frac{L}{\mu}$ is called the condition number of f.
  - For least squares, it's the "matrix condition number" of $\nabla^2 f(w)$.

Summary

- Guaranteed progress bound if gradient is Lipschitz, based on norm of gradient.
- Practical step size strategies based on the progress bound.
- Error on iteration t of $O(1/t)$ for functions that are bounded below.
  - Implies that we need $t = O(1/\epsilon)$ iterations to have $\|\nabla f(x^k)\| \leq \epsilon$.
- Polyak-Łojasiewicz inequality leads to linear convergence of gradient descent.
  - Only needs $O(\log(1/\epsilon))$ iterations to get within ϵ of global optimum.
- Strongly-convex differentiable functions functions satisfy PL-inequality.
  - Adding L2-regularization makes gradient descent go faster.

Checking Derivative Code

- Gradient descent codes require you to write objective/gradient code.
  - This tends to be error-prone, although automatic differentiation codes are helping.

- Make sure to check your derivative code:
  - Numerical approximation to partial derivative:

$$\nabla_i f(x) \approx \frac{f(x + \delta e_i) - f(x)}{\delta}$$

  - For large-scale problems you can check a random direction d:

$$\nabla f(x)^T d \approx \frac{f(x + \delta d) - f(x)}{\delta}$$

  - If the left side coming from your code is very different from the right side, there is likely a bug.

Lipschitz Continuity of Logistic Regression Gradient

- Logistic regression Hessian is

$$\nabla^2 f(w) = \sum_{i=1}^{n} \underbrace{h(y_i w^T x^i) h(-y^i w^T x^i)}_{d_{ii}} x^i (x^i)^T$$

$$\preceq 0.25 \sum_{i=1}^{n} x^i (x^i)^T$$

$$= 0.25 X^T X.$$

- In the second line we use that $h(\alpha) \in (0,1)$ and $h(-\alpha) = 1 - \alpha$.
  - This means that $d_{ii} \leq 0.25$.

- So for logistic regression, we can take $L = \frac{1}{4} \max\{\text{eig}(X^T X)\}$.

Why the gradient descent iteration?

- For a C^2 function, a variation on the multivariate Taylor expansion is that

$$f(v) = f(w) + \nabla f(w)^T(v-w) + \frac{1}{2}(v-w)^T\nabla^2 f(u)(v-w),$$

for any w and v (with u being some convex combination of w and v).

- If w and v are very close to each other, then we have

$$f(v) = f(w) + \nabla f(w)^T(v-w) + O(\|v-w\|^2),$$

and the last term becomes negligible.

- Ignoring the last term, for a fixed $\|v-w\|$ I can minimize $f(v)$ by choosing $(v-w) \propto -\nabla f(w)$.
    - So if we're moving a small amount the optimal choice is gradient descent.

Descent Lemma for C^1 Functions

- Let ∇f be L-Lipschitz continuous, and define $g(\alpha) = f(x + \alpha z)$ for a scalar α.

$$f(y) = f(x) + \int_0^1 \nabla f(x + \alpha(y-x))^T(y-x)d\alpha \quad \text{(fund. thm. calc.)}$$

$$\text{($\pm$ const.)} = f(x) + \nabla f(x)^T(y-x) + \int_0^1 (\nabla f(x + \alpha(y-x)) - \nabla f(x))^T (y-x)d\alpha$$

$$\text{(CS ineq.)} \leq f(x) + \nabla f(x)^T(y-x) + \int_0^1 \|\nabla f(x + \alpha(y-x)) - \nabla f(x)\|\|y-x\|d\alpha$$

$$\text{(Lipschitz)} \leq f(x) + \nabla f(x)^T(y-x) + \int_0^1 L\|x + \alpha(y-x) - x\|\|y-x\|d\alpha$$

$$\text{(homog.)} = f(x) + \nabla f(x)^T(y-x) + \int_0^1 L\alpha\|y-x\|^2 d\alpha$$

$$(\int_0^1 \alpha = \frac{1}{2}) = f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}\|y-x\|^2.$$

Equivalent Conditions to Lipschitz Continuity of Gradient

- We said that Lipschitz continuity of the gradient

$$\|\nabla f(w) - \nabla f(v)\| \leq L\|w - v\|,$$

  is equivalent for C^2 functions to having

$$\nabla^2 f(w) \preceq LI.$$

- There are a lot of other equivalent definitions, see here:
  - `http://xingyuzhou.org/blog/notes/Lipschitz-gradient`.

Why is $\mu \leq L$?

- The descent lemma for functions with L-Lipschitz ∇f is that

$$f(v) \leq f(w) + \nabla f(w)^\top (v - w) + \frac{L}{2}\|v - w\|^2.$$

- Minimizing both sides in terms of v (by taking the gradient and setting to 0 and observing that it's convex) gives

$$f^* \leq f(w) - \frac{1}{2L}\|\nabla f(w)\|^2.$$

- So with PL and Lipschitz we have

$$\frac{1}{2\mu}\|\nabla f(w)\|^2 \geq f(w) - f^* \geq \frac{1}{2L}\|\nabla f(w)\|^2,$$

which implies $\mu \leq L$.

Strong Convexity Implies PL Inequality

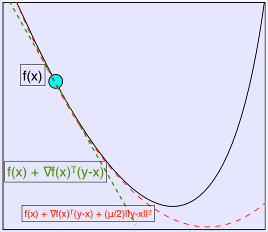
- As before, from Taylor's theorem we have for C^2 functions that

$$f(v) = f(w) + \nabla f(w)^\top (v - w) + \frac{1}{2}(v - w)^\top \nabla^2 f(u)(v - w).$$

- By strong-convexity, $d^\top \nabla^2 f(u) d \geq \mu \|d\|^2$ for any d and u.

$$f(v) \geq f(w) + \nabla f(w)^\top (v - w) + \frac{\mu}{2}\|v - w\|^2$$

- Treating right side as function of v, we get a quadratic lower bound on f.



Strong Convexity Implies PL Inequality

- As before, from Taylor's theorem we have for C^2 functions that

$$f(v) = f(w) + \nabla f(w)^\top (v - w) + \frac{1}{2}(v - w)^\top \nabla^2 f(u)(v - w).$$

- By strong-convexity, $d^\top \nabla^2 f(u) d \geq \mu \|d\|^2$ for any d and u.

$$f(v) \geq f(w) + \nabla f(w)^\top (v - w) + \frac{\mu}{2}\|v - w\|^2.$$

- Treating right side as function of v, we get a quadratic lower bound on f.

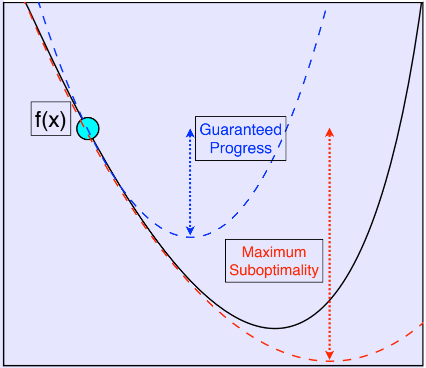
- Minimize both sides in terms of v gives

$$f^* \geq f(w) - \frac{1}{2\mu}\|\nabla f(w)\|^2,$$

which is the PL inequality (bonus slides show for C^1 functions).

Combining Lipschitz Continuity and Strong Convexity

- Lipschitz continuity of gradient gives guaranteed progress.
- Strong convexity of functions gives maximum sub-optimality.



- Progress on each iteration will be at least a fixed fraction of the sub-optimality.

C^1 Strongly-Convex Functions satisfy PL

- If $g(x) = f(x) - \frac{\mu}{2}\|x\|^2$ is convex then from C^1 definition of convexity

$$g(y) \geq g(x) + \nabla g(x)^\top (y - x)$$

or that

$$f(y) - \frac{\mu}{2}\|y\|^2 \geq f(x) - \frac{\mu}{2}\|x\|^2 + (\nabla f(x) - \mu x)^\top (y - x),$$

which gives

$$\begin{aligned} f(y) &\geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y\|^2 - \mu x^\top y + \frac{\mu}{2}\|x\|^2 \\ &= f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|^2, \quad \text{(complete square)} \end{aligned}$$

the inequality we used to show C^2 strongly-convex function f satisfies PL.

PL Inequality for Least Squares

- Least squares can be written as $f(x) = g(Ax)$ for a σ-strongly-convex g and matrix A, we'll show that the PL inequality is satisfied for this type of function.
- The function is minimized at some $f(y^*)$ with $y^* = Ax$ for some x, let's use $\mathcal{X}^* = \{x | Ax = y^*\}$ as the set of minimizers. We'll use x_p as the "projection" (defined next lecture) of x onto $\mathcal{X}^*$.

$$
\begin{aligned}
f^* = f(x_p) &\geq f(x) + \langle \nabla f(x), x_p - x \rangle + \frac{\sigma}{2} ||A(x_p - x)||^2 \\
&\geq f(x) + \langle \nabla f(x), x_p - x \rangle + \frac{\sigma \theta(A)}{2} ||x_p - x||^2 \\
&\geq f(x) + \min_y \left[\langle \nabla f(x), y - x \rangle + \frac{\sigma \theta(A)}{2} ||y - x||^2 \right] \\
&= f(x) - \frac{1}{2\theta(A)\sigma} ||\nabla f(x)||^2.
\end{aligned}
$$

- The first line uses strong-convexity of g, the second line uses the "Hoffman bound" which relies on $\mathcal{X}^*$ being a polyhedral set defined in this particular way to give a constant $\theta(A)$ depending on A that holds for all x (in this case it's the smallest non-zero singular value of A), and the third line uses that x_p is a particular y in the min.

Linear Convergence for "Locally-Nice" Functions

- For linear convergence it's sufficient to have

$$L[f(x^{t+1}) - f(x^t)] \geq \frac{1}{2}\|\nabla f(x^t)\|^2 \geq \mu[f(x^t) - f^*],$$

  for all x^t for some L and μ with $L \geq \mu > 0$.

  (technically, we could even get rid of the connection to the gradient)

- Notice that this only needs to hold for all x^t, not for all possible x.
  - We could get linear rate for "nasty" function if the iterations stay in a "nice" region.
  - We can get lucky and converge faster than the global L/μ would suggest.

- Arguments like this give linear rates for some non-convex problems like PCA.

Convergence of Iterates

- Under strong-convexity, you can also show that the iterations converge linearly.
- With a step-size of $1/L$ you can show that

$$\|w^{k+1} - w^*\| \leq \left(1 - \frac{\mu}{L}\right)\|w^k - w^*\|.$$

- If you use a step-size of $2/(\mu + L)$ this improves to

$$\|w^{k+1} - w^*\| \leq \left(\frac{L-\mu}{L+\mu}\right)\|w^k - w^*\|.$$

- Under PL, the solution w^* is not unique.
  - You can show linear convergence of $\|w^k - w_p^k\|$, where w_p^k is closest solution.