Gradient descent revisited

Geoff Gordon & Ryan Tibshirani
Optimization 10-725 / 36-725

Gradient descent

Recall that we have $f : \mathbb{R}^n \to \mathbb{R}$, convex and differentiable, want to solve

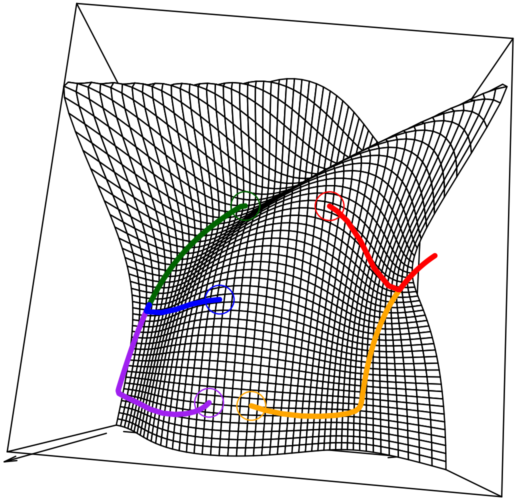
$$\min_{x \in \mathbb{R}^n} \; f(x),$$

i.e., find $x^\star$ such that $f(x^\star) = \min f(x)$

Gradient descent: choose initial $x^{(0)} \in \mathbb{R}^n$, repeat:

$$x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

Stop at some point



Interpretation

At each iteration, consider the expansion

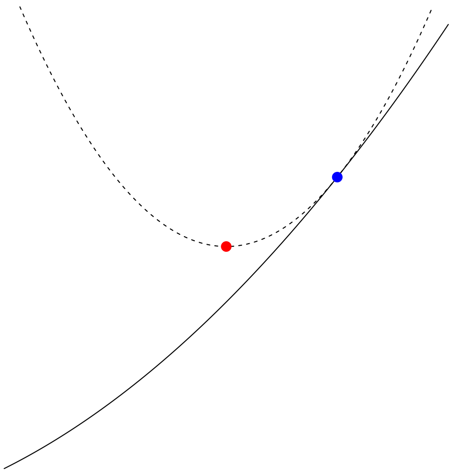
$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t} \|y - x\|^2$$

Quadratic approximation, replacing usual $\nabla^2 f(x)$ by $\frac{1}{t} I$

$f(x) + \nabla f(x)^T (y - x)$	linear approximation to f
$\frac{1}{2t} \|y - x\|^2$	proximity term to x, with weight $1/(2t)$

Choose next point $y = x^+$ to minimize quadratic approximation

$$x^+ = x - t \nabla f(x)$$



Blue point is x, red point is x^+

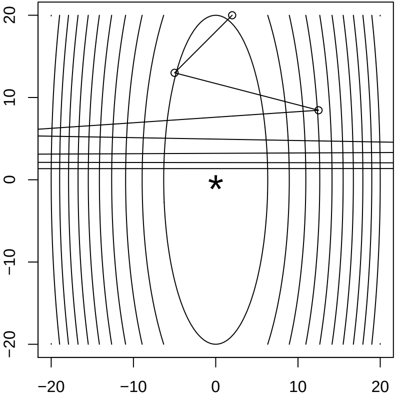
Outline

Today:

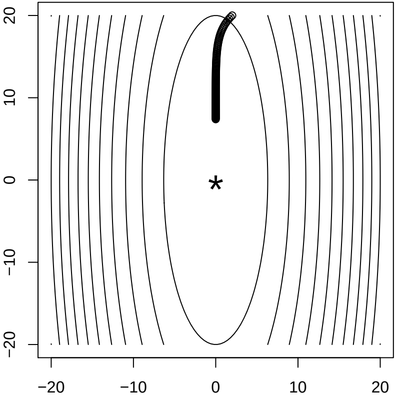
- How to choose step size t_k
- Convergence under Lipschitz gradient
- Convergence under strong convexity
- Forward stagewise regression, boosting

Fixed step size

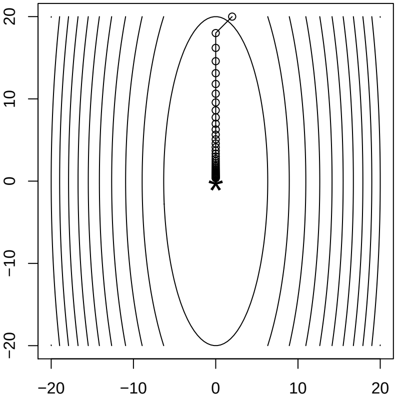
Simply take $t_k = t$ for all $k = 1, 2, 3, \ldots$, can diverge if t is too big. Consider $f(x) = (10x_1^2 + x_2^2)/2$, gradient descent after 8 steps:



Can be slow if t is too small. Same example, gradient descent after 100 steps:



Same example, gradient descent after 40 appropriately sized steps:



This porridge is too hot! – too cold! – juuussst right. Convergence analysis later will give us a better idea

Backtracking line search

A way to adaptively choose the step size

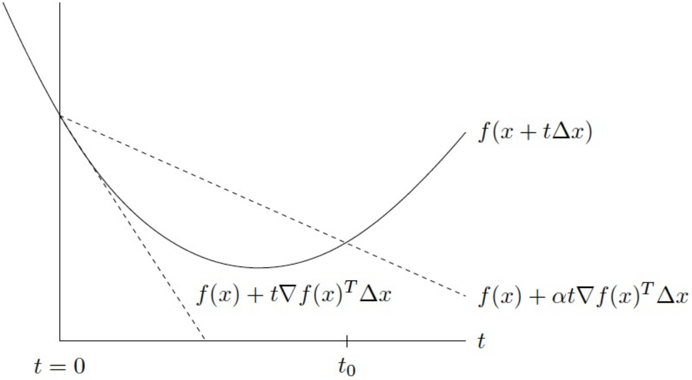
- First fix a parameter $0 < \beta < 1$
- Then at each iteration, start with $t = 1$, and while

$$f(x - t\nabla f(x)) > f(x) - \frac{t}{2}\|\nabla f(x)\|^2,$$

  update $t = \beta t$

Simple and tends to work pretty well in practice

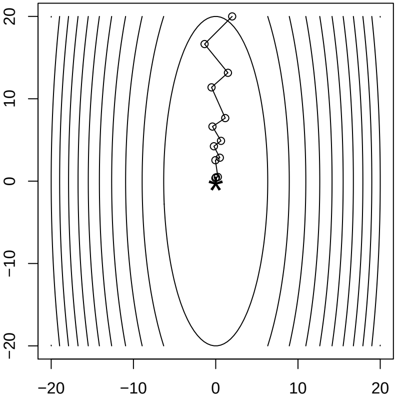
Interpretation



(From B & V page 465)

For us $\Delta x = -\nabla f(x)$, $\alpha = 1/2$

Backtracking picks up roughly the right step size (13 steps):



Here $\beta = 0.8$ (B & V recommend $\beta \in (0.1, 0.8)$)

Exact line search

At each iteration, do the best we can along the direction of the gradient,

$$t = \underset{s \geq 0}{\operatorname{argmin}} \; f(x - s\nabla f(x))$$

Usually not possible to do this minimization exactly

Approximations to exact line search are often not much more efficient than backtracking, and it's not worth it

Convergence analysis

Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and additionally

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \text{for any } x, y$$

I.e., ∇f is Lipschitz continuous with constant $L > 0$

Theorem: Gradient descent with fixed step size $t \leq 1/L$ satisfies

$$f(x^{(k)}) - f(x^\star) \leq \frac{\|x^{(0)} - x^\star\|^2}{2tk}$$

I.e., gradient descent has convergence rate $O(1/k)$

I.e., to get $f(x^{(k)}) - f(x^\star) \leq \epsilon$, need $O(1/\epsilon)$ iterations

Proof

Key steps:

- ∇f Lipschitz with constant $L \Rightarrow$

$$f(y) \le f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}\|y - x\|^2 \quad \text{all } x, y$$

- Plugging in $y = x - t\nabla f(x)$,

$$f(y) \le f(x) - (1 - \frac{Lt}{2}) t \|\nabla f(x)\|^2$$

- Letting $x^+ = x - t\nabla f(x)$ and taking $0 < t \le 1/L$,

$$\begin{aligned} f(x^+) &\le f(x^\star) + \nabla f(x)^T (x - x^\star) - \frac{t}{2}\|\nabla f(x)\|^2 \\ &= f(x^\star) + \frac{1}{2t}\big(\|x - x^\star\|^2 - \|x^+ - x^\star\|^2\big) \end{aligned}$$

- Summing over iterations:

$$\sum_{i=1}^{k}\left(f(x^{(i)}) - f(x^\star)\right) \le \frac{1}{2t}\left(\|x^{(0)} - x^\star\|^2 - \|x^{(k)} - x^\star\|^2\right)$$
$$\le \frac{1}{2t}\|x^{(0)} - x^\star\|^2$$

- Since $f(x^{(k)})$ is nonincreasing,

$$f(x^{(k)}) - f(x^\star) \le \frac{1}{k}\sum_{i=1}^{k}\left(f(x^{(i)}) - f(x^\star)\right) \le \frac{\|x^{(0)} - x^\star\|^2}{2tk}$$

$\square$

Convergence analysis for backtracking

Same assumptions, $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable, and ∇f is Lipschitz continuous with constant $L > 0$

Same rate for a step size chosen by backtracking search

Theorem: Gradient descent with backtracking line search satisfies

$$f(x^{(k)}) - f(x^\star) \le \frac{\|x^{(0)} - x^\star\|^2}{2t_{\mathsf{min}}k}$$

where $t_{\mathsf{min}} = \min\{1, \beta/L\}$

If β is not too small, then we don't lose much compared to fixed step size (β/L vs $1/L$)

Strong convexity

Strong convexity of f means for some $d > 0$,

$$\nabla^2 f(x) \succeq dI \quad \text{for any } x$$

Better lower bound than that from usual convexity:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{d}{2}\|y - x\|^2 \quad \text{all } x, y$$

Under Lipschitz assumption as before, and also strong convexity:

Theorem: Gradient descent with fixed step size $t \leq 2/(d + L)$ or with backtracking line search search satisfies

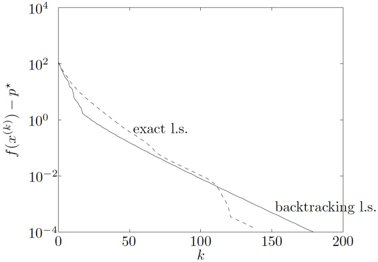
$$f(x^{(k)}) - f(x^\star) \leq c^k \frac{L}{2}\|x^{(0)} - x^\star\|^2$$

where $0 < c < 1$

I.e., rate with strong convexity is $O(c^k)$, exponentially fast!

I.e., to get $f(x^{(k)}) - f(x^\star) \leq \epsilon$, need $O(\log(1/\epsilon))$ iterations

Called linear convergence, because looks linear on a semi-log plot:



(From B & V page 487)

Constant c depends adversely on condition number L/d (higher condition number $\Rightarrow$ slower rate)

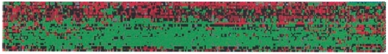
How realistic are these conditions?

How realistic is Lipschitz continuity of ∇f?

- This means $\nabla^2 f(x) \preceq LI$
- E.g., consider $f(x) = \frac{1}{2}\|y - Ax\|^2$ (linear regression). Here $\nabla^2 f(x) = A^T A$, so ∇f Lipschitz with $L = \sigma^2_{\mathsf{max}}(A) = \|A\|^2$

How realistic is strong convexity of f?

- Recall this is $\nabla^2 f(x) \succeq dI$
- E.g., again consider $f(x) = \frac{1}{2}\|y - Ax\|^2$, so $\nabla^2 f(x) = A^T A$, and we need $d = \sigma^2_{\mathsf{min}}(A)$
- If A is wide, then $\sigma_{\mathsf{min}}(A) = 0$, and f can't be strongly convex

  (E.g., $A =$ )
- Even if $\sigma_{\mathsf{min}}(A) > 0$, can have a very large condition number $L/d = \sigma_{\mathsf{max}}(A)/\sigma_{\mathsf{min}}(A)$

Practicalities

Stopping rule: stop when $\|\nabla f(x)\|$ is small

- Recall $\nabla f(x^\star) = 0$
- If f is strongly convex with parameter d, then

$$\|\nabla f(x)\| \leq \sqrt{2d\epsilon} \;\Rightarrow\; f(x) - f(x^\star) \leq \epsilon$$

Pros and cons:

- Pro: simple idea, and each iteration is cheap
- Pro: Very fast for well-conditioned, strongly convex problems
- Con: Often slow, because interesting problems aren't strongly convex or well-conditioned
- Con: can't handle nondifferentiable functions

Forward stagewise regression

Let's stick with $f(x) = \frac{1}{2}\|y - Ax\|^2$, linear regression

A is $n \times p$, its columns $A_1, \dots A_p$ are predictor variables

Forward stagewise regression: start with $x^{(0)} = 0$, repeat:

- Find variable i such that $|A_i^T r|$ is largest, for $r = y - Ax^{(k-1)}$ (largest absolute correlation with residual)
- Update $x_i^{(k)} = x_i^{(k-1)} + \gamma \cdot \mathrm{sign}(A_i^T r)$

Here $\gamma > 0$ is small and fixed, called learning rate

This looks kind of like gradient descent

Steepest descent

Close cousin to gradient descent, just change the choice of norm. Let q, r be complementary (dual): $1/q + 1/r = 1$

Updates are $x^+ = x + t \cdot \Delta x$, where

$$\Delta x = \|\nabla f(x)\|_r \cdot u$$
$$u = \operatorname*{argmin}_{\|v\|_q \le 1} \nabla f(x)^T v$$

- If $q = 2$, then $\Delta x = -\nabla f(x)$, gradient descent
- If $q = 1$, then $\Delta x = -\partial f(x)/\partial x_i \cdot e_i$, where

$$\left|\frac{\partial f}{\partial x_i}(x)\right| = \max_{j=1,\dots n} \left|\frac{\partial f}{\partial x_j}(x)\right| = \|\nabla f(x)\|_\infty$$

Normalized steepest descent just takes $\Delta x = u$ (unit q-norm)

Equivalence

Normalized steepest descent with 1-norm: updates are

$$x_i^+ = x_i - t \cdot \mathrm{sign}\Big\{\frac{\partial f}{\partial x_i}(x)\Big\}$$

where i is the largest component of $\nabla f(x)$ in absolute value

Compare forward stagewise: updates are

$$x_i^+ = x_i + \gamma \cdot \mathrm{sign}(A_i^T r), \quad r = y - Ax$$

Recall here $f(x) = \frac{1}{2}\|y - Ax\|^2$, so $\nabla f(x) = -A^T(y - Ax)$ and $\partial f(x)/\partial x_i = -A_i^T(y - Ax)$

Forward stagewise regression is exactly normalized steepest descent under 1-norm

Early stopping and regularization

Forward stagewise is like a slower version of forward stepwise

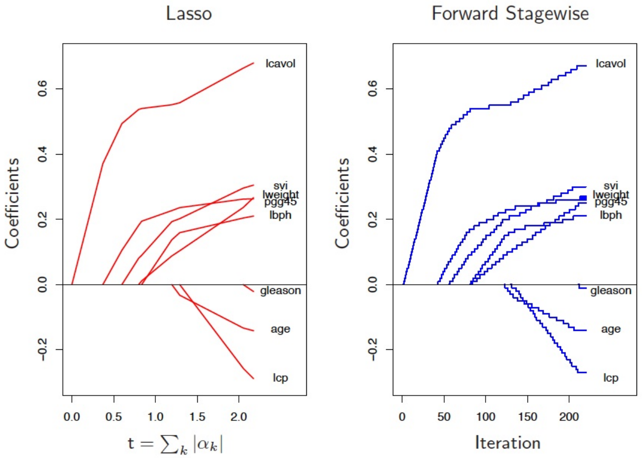
If we stop early, i.e., don't continue all the way to the least squares solution, then we get a sparse approximation ... can this be used as a form of regularization?

Recall lasso problem:

$$\min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - Ax\|^2 \ \text{subject to} \ \|x\|_1 \le t$$

Solution $x^\star(s)$, as function of s, also exhibits varying amounts of regularization

How do they compare?



(From ESL page 609)

For some problems (some y, A), with a small enough step size, forward stagewise iterates trace out lasso solution path!

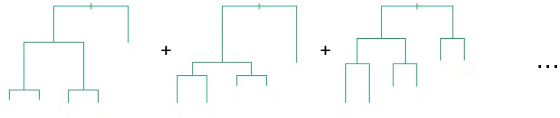
Gradient boosting

Given observations $y = (y_1, \ldots y_n) \in \mathbb{R}^n$, associated predictor measurements $a_i \in \mathbb{R}^p$, $i = 1, \ldots n$

Want to construct a flexible (nonlinear) model for y based on predictors. Weighted sum of trees:

$$y_i \approx \hat{y}_i = \sum_{j=1}^{m} \gamma_j \cdot T_j(a_i), \quad i = 1, \ldots n$$

Each tree T_j inputs predictor measurements a_i, outputs prediction. Trees are typically very short



Pick a loss function L that reflects task; e.g., for continuous y, could take $L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$

Want to solve

$$\min_{\gamma \in \mathbb{R}^M} \sum_{i=1}^{n} L\Big(y_i, \sum_{j=1}^{M} \gamma_j \cdot T_j(a_i)\Big)$$

Indexes all trees of a fixed size (e.g., depth $= 5$), so M is huge

Space is simply too big to optimize

Gradient boosting: combines gradient descent idea with forward model building

First think of minimization as $\min f(\hat{y})$, function of predictions $\hat{y}$ (subject to $\hat{y}$ coming from trees)

Start with initial model, i.e., fit a single tree $\hat{y}^{(0)} = T_0$. Repeat:

- Evaluate gradient g at latest prediction $\hat{y}^{(k-1)}$,

$$g_i = \left[\frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i} \right] \Bigg|_{\hat{y}_i = \hat{y}_i^{(k-1)}}, \quad i = 1, \ldots n$$

- Find a tree T_k that is close to $-g$, i.e., T_k solves

$$\min_T \sum_{i=1}^{n} (-g_i - T(a_i))^2$$

  Not hard to (approximately) solve for a single tree

- Update our prediction:

$$\hat{y}^{(k)} = \hat{y}^{(k-1)} + \gamma_k \cdot T_k$$

  Note: predictions are weighted sums of trees, as desired!

Lower bound

Remember $O(1/k)$ rate for gradient descent over problem class: convex, differentiable functions with Lipschitz continuous gradients

First-order method: iterative method, updates $x^{(k)}$ in

$$x^{(0)} + \text{span}\{\nabla f(x^{(0)}), \nabla f(x^{(1)}), \dots \nabla f(x^{(k-1)})\}$$

Theorem (Nesterov): For any $k \leq (n-1)/2$ and any starting point $x^{(0)}$, there is a function f in the problem class such that any first-order method satisfies

$$f(x^{(k)}) - f(x^\star) \geq \frac{3L\|x^{(0)} - x^\star\|^2}{32(k+1)^2}$$

Can we achieve a rate $O(1/k^2)$? Answer: yes, and more!

References

- S. Boyd and L. Vandenberghe (2004), *Convex Optimization*, Cambridge University Press, Chapter 9
- T. Hastie, R. Tibshirani and J. Friedman (2009), *The Elements of Statistical Learning*, Springer, Chapters 10 and 16
- Y. Nesterov (2004), *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer Academic Publishers, Chapter 2
- L. Vandenberghe, Lecture Notes for EE 236C, UCLA, Spring 2011-2012