Gradient descent

Barnabas Poczos & Ryan Tibshirani
Convex Optimization 10-725/36-725

Gradient descent

First consider unconstrained minimization of $f : \mathbb{R}^n \to \mathbb{R}$, convex and differentiable. We want to solve

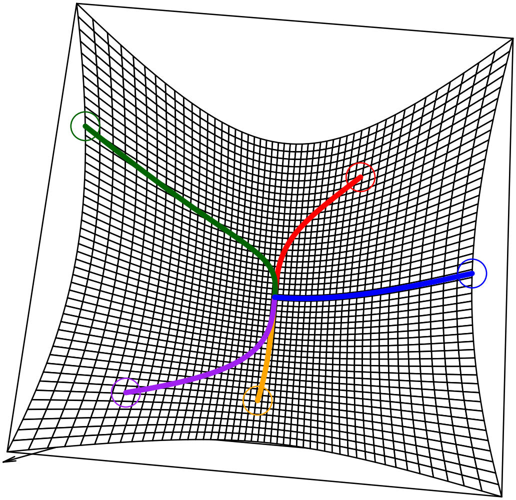
$$\min_{x \in \mathbb{R}^n} \; f(x),$$

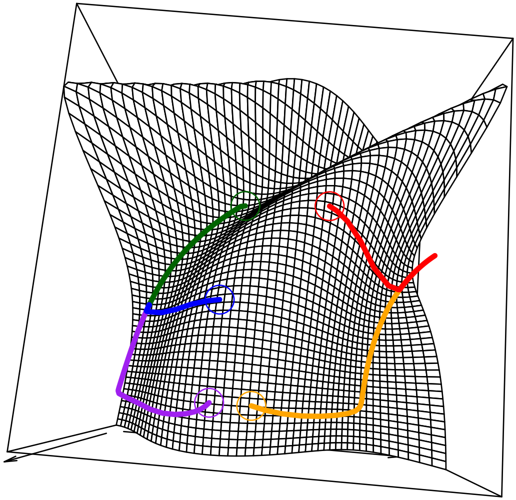
i.e., find $x^\star$ such that $f(x^\star) = \min_x f(x)$

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

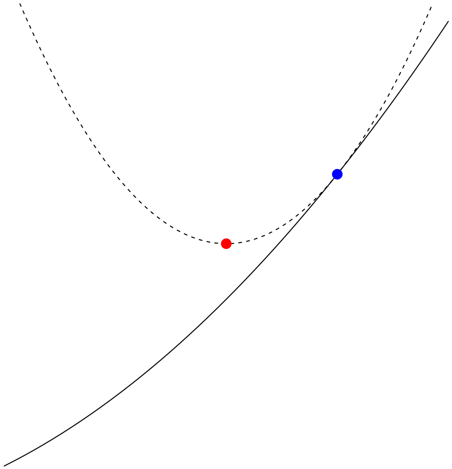
$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t}\|y - x\|_2^2$$

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
$\frac{1}{2t}\|y - x\|_2^2$	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

$$x^+ = \text{argmin}_{y \in \mathbb{R}^n} \; f(x) + \nabla f(x)^T (y-x) + \|y-x\|_2^2/(2t)$$

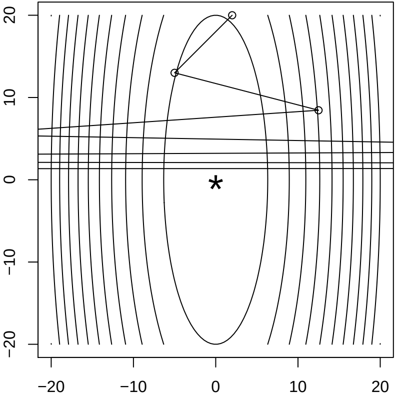
Outline

Today:

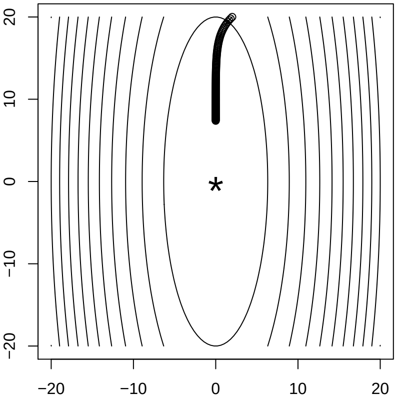
- How to choose step size t_k
- Convergence under Lipschitz gradient
- Convergence under strong convexity
- Forward stagewise regression, boosting

Fixed step size

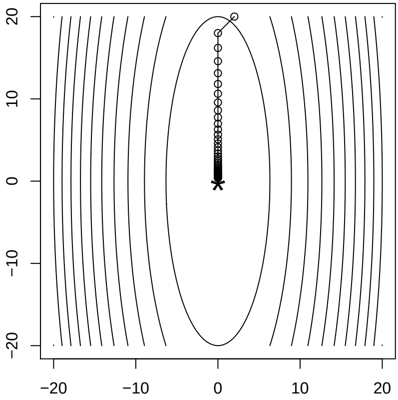
Simply take $t_k = t$ for all $k = 1, 2, 3, \ldots$, can diverge if t is too big. Consider $f(x) = (10x_1^2 + x_2^2)/2$, gradient descent after 8 steps:



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

One way to adaptively choose the step size is to use backtracking line search:

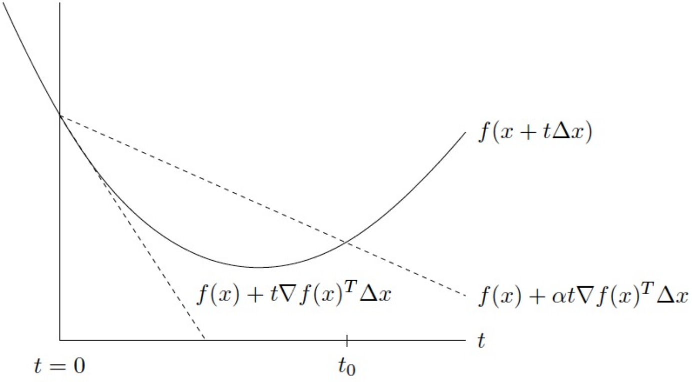
- First fix parameters $0 < \beta < 1$ and $0 < \alpha \leq 1/2$
- Then at each iteration, start with $t = 1$, and while

$$f(x - t\nabla f(x)) > f(x) - \alpha t \|\nabla f(x)\|_2^2,$$

  update $t = \beta t$

Simple and tends to work pretty well in practice

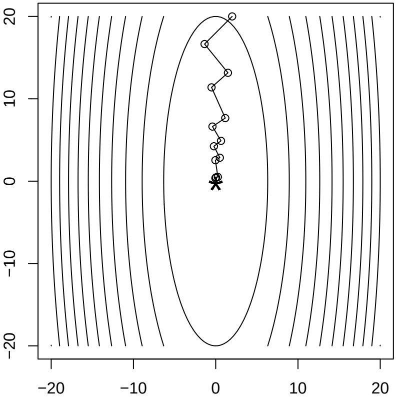
Interpretation



(From B & V page 465)

For us $\Delta x = -\nabla f(x)$

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

Could also choose step to do the best we can along the direction of the negative gradient, called exact line search:

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

Usually not possible to do this minimization exactly

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

Convergence analysis

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

$$\|\nabla f(x) - \nabla f(y)\|_2 \le L\|x - y\|_2 \quad \text{for any } x, y$$

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

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

$$f(x^{(k)}) - f(x^\star) \le \frac{\|x^{(0)} - x^\star\|_2^2}{2tk}$$

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

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

Proof

Key steps:

- ∇f Lipschitz with constant $L \Rightarrow$

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}\|y - x\|_2^2 \quad \text{all } x, y$$

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

$$f(x^+) \leq f(x) - (1 - \frac{Lt}{2})t\|\nabla f(x)\|_2^2$$

- Taking $0 < t \leq 1/L$, and using convexity of f,

$$\begin{aligned} f(x^+) &\leq f(x^\star) + \nabla f(x)^T (x - x^\star) - \frac{t}{2}\|\nabla f(x)\|_2^2 \\ &= f(x^\star) + \frac{1}{2t}\left(\|x - x^\star\|_2^2 - \|x^+ - x^\star\|_2^2\right) \end{aligned}$$

- Summing over iterations:

$$\sum_{i=1}^{k}(f(x^{(i)}) - f(x^\star)) \leq \frac{1}{2t}\left(\|x^{(0)} - x^\star\|_2^2 - \|x^{(k)} - x^\star\|_2^2\right)$$
$$\leq \frac{1}{2t}\|x^{(0)} - x^\star\|_2^2$$

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

$$f(x^{(k)}) - f(x^\star) \leq \frac{1}{k}\sum_{i=1}^{k}\left(f(x^{(i)}) - f(x^\star)\right) \leq \frac{\|x^{(0)} - x^\star\|_2^2}{2tk}$$

□

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

$$f(x^{(k)}) - f(x^\star) \le \frac{\|x^{(0)} - x^\star\|_2^2}{2t_{\mathsf{min}}k}$$

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 $d > 0$,

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

Sharper lower bound than that from usual convexity:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{d}{2} \|y - x\|_2^2 \quad \text{all } x, y$$

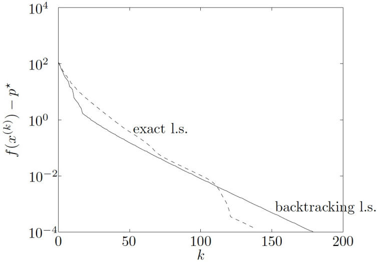
Under Lipschitz assumption as before, and also strong convexity:

> **Theorem:** Gradient descent with fixed step size $t \leq 2/(d + L)$ or with backtracking line search search satisfies
>
> $$f(x^{(k)}) - f(x^\star) \leq c^k \frac{L}{2} \|x^{(0)} - x^\star\|_2^2$$
>
> where $0 < c < 1$

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

I.e., to get $f(x^{(k)}) - f(x^\star) \le \epsilon$, 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)

A look at the conditions

Lipschitz continuity of ∇f:

- This means $\nabla^2 f(x) \preceq LI$
- E.g., consider $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ (linear regression). Here $\nabla^2 f(\beta) = X^T X$, so ∇f is Lipschitz with $L = \sigma^2_{\mathrm{max}}(X)$

Strong convexity of f:

- Recall this is $\nabla^2 f(x) \succeq dI$
- E.g., consider $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$, with $\nabla^2 f(\beta) = X^T X$. Now we need $d = \sigma^2_{\mathrm{min}}(X)$
- If X is wide—i.e., X is $n \times p$ with $p > n$—then $\sigma_{\mathrm{min}}(X) = 0$, and f can't be strongly convex
- Even if $\sigma_{\mathrm{min}}(X) > 0$, can have a very large condition number $L/d = \sigma_{\mathrm{max}}(X)/\sigma_{\mathrm{min}}(X)$

A function f having Lipschitz gradient and being strongly convex can be summarized as:

$$dI \preceq \nabla^2 f(x) \preceq LI \quad \text{for all } x \in \mathbb{R}^n,$$

for constants $L > d > 0$

Think of f being sandwiched between two quadratics

This may seem like a strong condition to hold globally, over all $x \in \mathbb{R}^n$. But a careful looks at the proofs shows we actually only need to have Lipschitz gradient and/or strong convexity over the sublevel set

$$S = \{x : f(x) \leq f(x^{(0)})\}$$

This is less restrictive

Practicalities

Stopping rule: stop when $\|\nabla f(x)\|_2$ is small

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

$$\|\nabla f(x)\|_2 \leq \sqrt{2d\epsilon} \;\Rightarrow\; f(x) - f(x^\star) \leq \epsilon$$

Pros and cons of gradient descent:

- 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 $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$, linear regression setting

X is $n \times p$, its columns $X_1, \ldots X_p$ are predictor variables

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

- Find variable i such that $|X_i^T r|$ is largest, where $r = y - X\beta^{(k-1)}$ (largest absolute correlation with residual)
- Update $\beta_i^{(k)} = \beta_i^{(k-1)} + \gamma \cdot \text{sign}(X_i^T r)$

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 p, q be complementary (dual): $1/p + 1/q = 1$

Steepest descent updates are $x^+ = x + t \cdot \Delta x$, where

$$\Delta x = \|\nabla f(x)\|_q \cdot u$$
$$u = \underset{\|v\|_p \le 1}{\text{argmin}} \; \nabla f(x)^T v$$

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

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

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

Equivalence

Normalized steepest descent with respect to ℓ_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

$$\beta_i^+ = \beta_i + \gamma \cdot \mathrm{sign}(X_i^T r), \quad r = y - X\beta$$

Recall here $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$, so $\nabla f(\beta) = -X^T(y - X\beta)$ and $\partial f(\beta)/\partial \beta_i = -X_i^T(y - X\beta)$

Forward stagewise regression is exactly normalized steepest descent under ℓ_1 norm (with fixed step size $t = \gamma$)

Early stopping and sparse approximation

If we run forward stagewise to completion, then we know that we will minimize the least squares criterion $f(\beta) = \|y - X\beta\|_2^2$, i.e., we will get a least squares solution

What happens if we stop early?

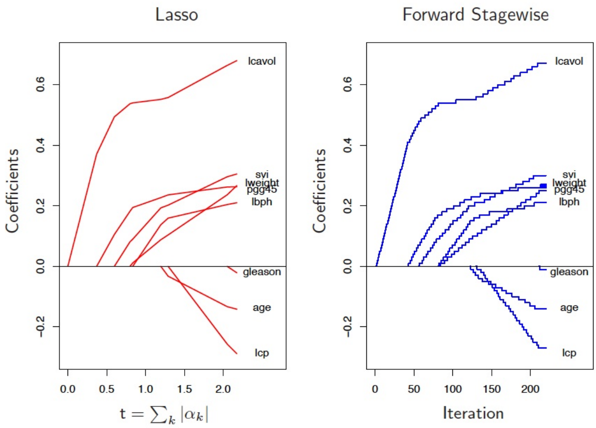
- May seem strange from an optimization perspective (we would be "under-optimizing") ...
- Interesting from a statistical perspective, because stopping early gives us a sparse approximation to the least squares solution

Well-known sparse regression estimator, the lasso:

$$\min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 \ \text{subject to} \ \|\beta\|_1 \le s$$

How do lasso solutions and forward stagewise estimates compare?

Left side is lasso solution $\hat{\beta}(s)$ over bound s, right side is forward stagewise estimate over iterations k:



(From ESL page 609)

For some problems, they are exactly the same (as $\gamma \to 0$)

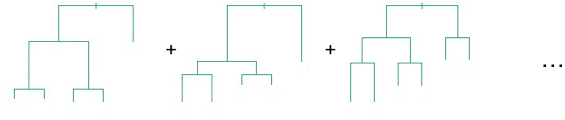
Gradient boosting

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

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

$$\hat{y}_i = \sum_{j=1}^{m} \beta_j \cdot T_j(x_i), \quad i = 1, \dots n$$

Each tree T_j inputs predictor measurements x_i, outputs prediction. Trees are grown typically pretty short



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

Want to solve

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

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

Space is simply too big to optimize

Gradient boosting: basically a version of gradient descent that's forced to work with trees

First think of minimization as $\min_{\hat{y}} f(\hat{y})$, function of predictions $\hat{y}$

Start with initial model, e.g., 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

$$\min_{\text{trees } T} \sum_{i=1}^{n} (-g_i - T(x_i))^2$$

  Not hard to (approximately) solve for a single tree

- Update our prediction:

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

  Note: predictions are weighted sums of trees, as desired

Can we do better?

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

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

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

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

$$f(x^{(k)}) - f(x^\star) \geq \frac{3L\|x^{(0)} - x^\star\|_2^2}{32(k+1)^2}$$

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

References

- S. Boyd and L. Vandenberghe (2004), "Convex optimization", Chapter 9
- T. Hastie, R. Tibshirani and J. Friedman (2009), "The elements of statistical learning", Chapters 10 and 16
- Y. Nesterov (2004), "Introductory lectures on convex optimization: a basic course", Chapter 2
- L. Vandenberghe, Lecture Notes for EE 236C, UCLA, Spring 2011-2012