Lecture 18: Newton's method vs. gradient descent

Ciaran Evans

Feedback summary

Thanks for feedback on the course! A brief summary of responses:

- Overall pace of the course – about right
- Overall workload – a bit high
  - Change to 2 projects, not 3
- HW 4 – too hard
- Deadlines – overwhelming preference for evening deadlines. I'll change that going forward
- Project 1: most people expected to use extension days. So:
  - Formal deadline moved to after Spring break
  - Ideally everyone is able to submit before break, but this gives you a few extra days if needed!

Previously

Gradient descent:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla f(\mathbf{x}^{(k)})$$

Newton's method:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k (\mathbf{H}_f(\mathbf{x}^{(k)}))^{-1} \nabla f(\mathbf{x}^{(k)})$$

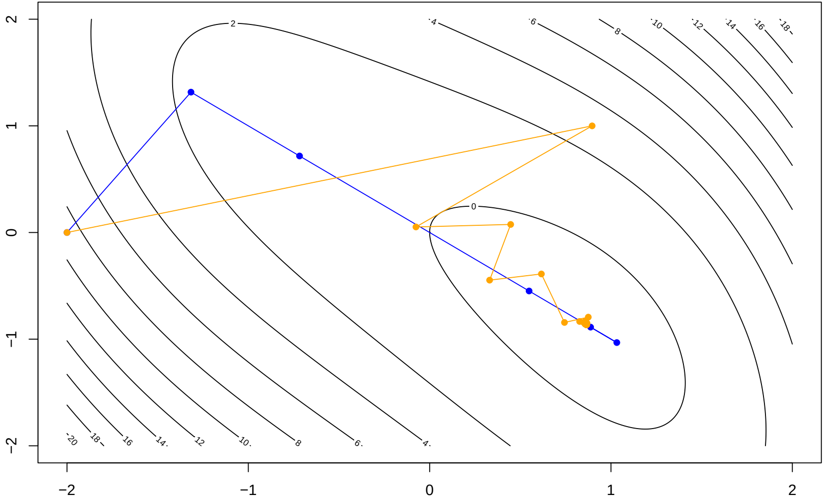
Today:

- Brief comparison between the two methods
- Which one gets used in practice?

Previously

Newton's method (blue): 7 iterations

Gradient descent with backtracking line search (orange): 27 iterations



Some properties of gradient descent

Question: What are some properties we have shown/observed about gradient descent?

Some properties of gradient descent

- The direction $\nabla f(\mathbf{x})$ is the direction of *steepest descent* (minimizes directional derivative)
- If α_k is chosen via exact line search, $\nabla f(\mathbf{x}^{(k+1)}) \perp \nabla f(\mathbf{x}^{(k)})$ (zig-zag pattern)
- Gradient descent takes many iterations in long, narrow valleys; scaling matters

Some properties of Newton's method

Gradient descent:

- The direction $-\nabla f(\mathbf{x})$ is the direction of *steepest descent* (minimizes directional derivative)
- If α_k is chosen via exact line search, $\nabla f(\mathbf{x}^{(k+1)}) \perp \nabla f(\mathbf{x}^{(k)})$ (zig-zag pattern)
- Gradient descent takes many iterations in long, narrow valleys; scaling matters

Newton's method:

- $-(\mathbf{H}_f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$ is a descent direction (directional derivative is negative)
- Not forced to take zig-zag steps
- Less susceptible to scaling issue

$-(\mathbf{H}_f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$ is a descent direction

Claim: Let $\mathbf{u}$ be a unit vector in the direction of $-(\mathbf{H}_f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$. Then $D_{\mathbf{u}}f(\mathbf{x}) < 0$ if $\mathbf{H}_f$ is a **positive definite** matrix

- **Definition (positive definite):** $\mathbf{H}_f(\mathbf{x})$ is a positive definite matrix if for all vectors $\mathbf{v} \neq 0$,

$$\mathbf{v}^T\mathbf{H}_f(\mathbf{x})\mathbf{v} > 0$$

- **Fact:** If $\mathbf{H}_f(\mathbf{x})$ is positive definite for all $\mathbf{x}$, then f is a **convex** function

$-(\mathbf{H}_f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$ is a descent direction

Claim: Let $\mathbf{u}$ be a unit vector in the direction of $-(\mathbf{H}_f(\mathbf{x}))^{-1}\nabla f(\mathbf{x})$. Then $D_{\mathbf{u}}f(\mathbf{x}) < 0$ if $\mathbf{H}_f$ is a **positive definite** matrix

- **Definition (positive definite):** $\mathbf{H}_f(\mathbf{x})$ is a positive definite matrix if for all vectors $\mathbf{v} \neq 0$,

$$\mathbf{v}^T\mathbf{H}_f(\mathbf{x})\mathbf{v} > 0$$

Proof of claim:

What actually gets used in practice? A broad generalization

Classical statistics: (parametric models with moderate size)
Newton's method

- Generalized linear models: Fisher scoring (Newton's method with Fisher info), often calculated with iteratively re-weighted least squares (IRLS)
- Generalized estimating equations
- Nonlinear least squares: Gauss-Newton (variant of Newton's method)

Modern statistical learning: (large models with *many* parameters)
Gradient descent

- Basic gradient descent and line search are not commonly used with large models
- Variations (stochastic gradient descent, momentum, subgradient methods, etc) are standard