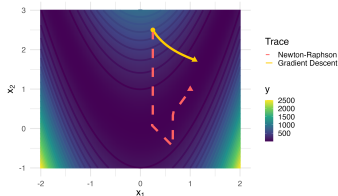
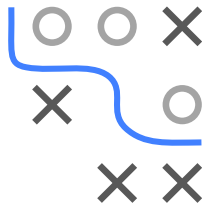


Optimization in Machine Learning

Second order methods

Newton-Raphson vs Gradient Descent



Learning goals

- Comparison of Newton-Raphson and Gradient Descent
- Pure Newton vs relaxed Newton with step size

NEWTON-RAPHSON AND GD (RECAP)

- Gradient Descent: **first order method**
⇒ *Gradient* information, i.e., first derivatives
- Newton-Raphson: **second order method**
⇒ *Hessian* information, i.e., second derivatives

Gradient Descent:

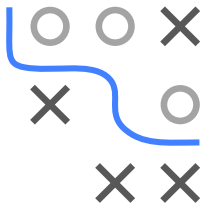
$$\theta^{[t+1]} = \theta^{[t]} - \alpha \nabla f(\theta^{[t]})$$

Pure Newton-Raphson:

$$\theta^{[t+1]} = \theta^{[t]} - (\nabla^2 f(\theta^{[t]}))^{-1} \nabla f(\theta^{[t]})$$

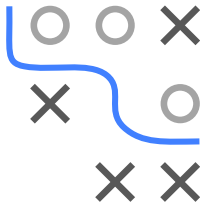
Relaxed/Damped Newton-Raphson:

$$\theta^{[t+1]} = \theta^{[t]} - \alpha (\nabla^2 f(\theta^{[t]}))^{-1} \nabla f(\theta^{[t]})$$



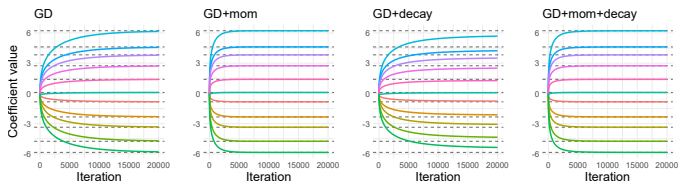
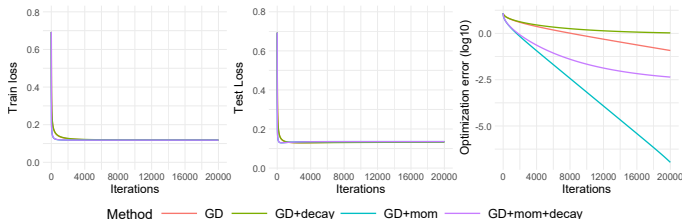
COMPARISON SIMULATION SET-UP

- Comparison of Newton-Raphson, relaxed NR and GD+momentum:
- **Logistic regression** (log loss) simulation with $n = 500$ samples and $p = 11$ features, where $\theta^* = (-5, -4, \dots, 0, \dots, 4, 5)^T$, and $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ for $\Sigma = \mathbf{I}$ (i.i.d.) or $\Sigma_{i,j} = 0.99^{|i-j|}$ (corr. features)
- To simulate response, we set $y^{(i)} \sim \mathcal{B}(\pi^{(i)})$, $\pi^{(i)} = \frac{1}{1 + e^{-(\mathbf{x}^{(i)})^T \theta^*}}$
- Indep. features result in a condition number of ≈ 2.9 while corr. features yield (moderately) bad condition number ≈ 600
- ERM has unique global minimizer (convexity) but no closed-form solution. We can approximate $\hat{\theta}$ using `glm` solution
- We also track the optimization error $\|\theta - \hat{\theta}\|_2$
- For relaxed NR we use $\alpha = 0.7$ and for GD we set $\alpha = 1$, momentum to 0.8 and use no step size control



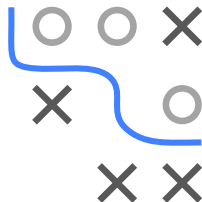
LOGISTIC REGRESSION (GD VARIANTS RECAP)

- Recall comparison of GD variants on log. reg. in last chapter:



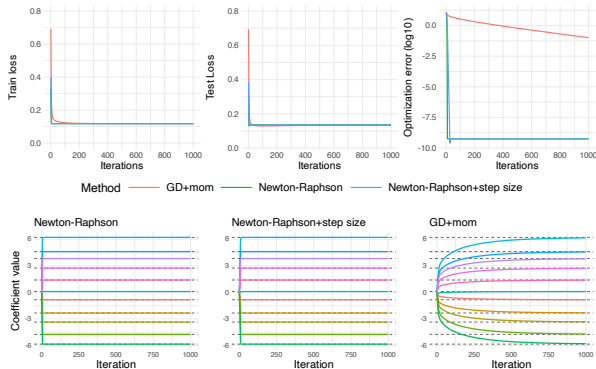
Dotted lines indicate global minimizers.

- GD+momentum** was fastest \Rightarrow now compare w/ Newton-Raphson
- NB:** GD+momentum only converges after several thousand steps



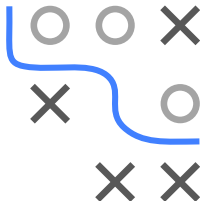
LOGISTIC REGRESSION (GD VS. NR)

- Let's run GD vs. NR for 1000 **steps** (independent features):



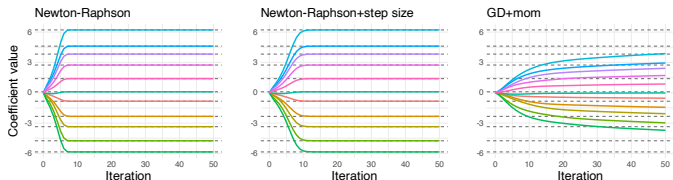
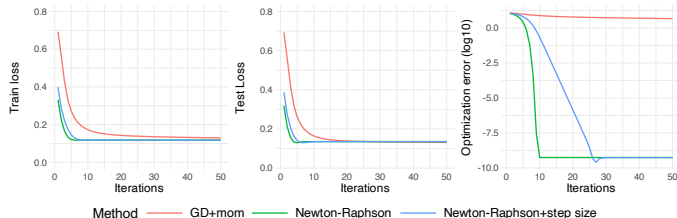
Dotted lines indicate global minimizers.

- NR** and **relaxed NR** \Rightarrow almost instantaneous convergence (see optimization error)
- Using $\alpha < 1$ slightly slows down **relaxed NR**
- GD+mom** several orders of magnitude slower than NR



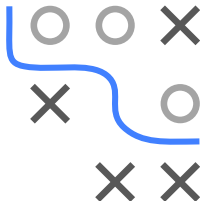
LOGISTIC REGRESSION (GD VS. NR)

- Let's run the same configuration only for 50 **steps** to see clearer picture:



Dotted lines indicate global minimizers.

- NR** takes ≈ 10 steps to reach same optimization error as **GD+mom** after 20,000 steps!
- Relaxed NR** with $\alpha < 1$ shows no advantage here



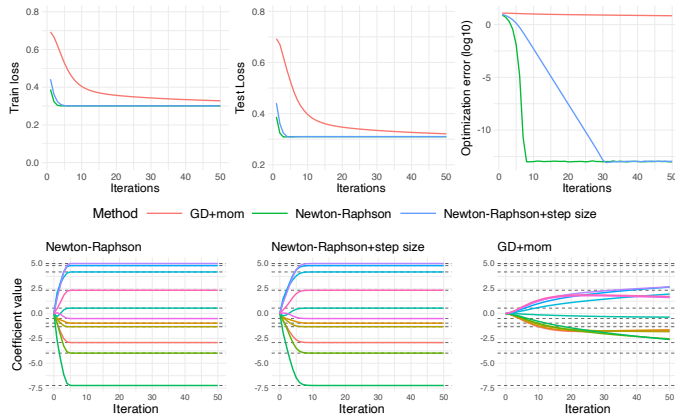
[illegible]

-
- The figure consists of two side-by-side line plots comparing three optimization methods: GD+mom (red line), Newton-Raphson (green line), and Newton-Raphson+step size (blue line). The x-axis for both plots is 'Seconds', ranging from 0.000 to 0.100.
- Left Plot: Training loss**
- The y-axis is 'Training loss', ranging from 0.0 to 0.6. All three methods start with a high training loss (around 0.6) at 0.000 seconds. The red line (GD+mom) drops sharply to near zero by 0.010 seconds. The green line (Newton-Raphson) drops to near zero by 0.015 seconds. The blue line (Newton-Raphson+step size) drops more gradually, reaching near zero by 0.025 seconds. All three methods maintain a near-zero training loss until 0.100 seconds.
- Right Plot: Parameter optimization error (log10)**
- The y-axis is 'Optimization error (log10)', ranging from -10.0 to 0.0. All three methods start with an optimization error of 0.0 at 0.000 seconds. The red line (GD+mom) drops sharply to approximately -9.5 by 0.010 seconds. The green line (Newton-Raphson) drops to approximately -9.5 by 0.015 seconds. The blue line (Newton-Raphson+step size) drops more gradually, reaching approximately -9.5 by 0.050 seconds. All three methods maintain a near-constant optimization error of approximately -9.5 until 0.100 seconds.
- Legend:**
- method — GD+mom (red line)
 - method — Newton-Raphson (green line)
 - method — Newton-Raphson+step size (blue line)

- **NR** steps are indeed slower than **GD** steps ($\approx 3\times$ here)
- But each NR step is so much better than GD ($\approx 2000\times$) that per-iteration runtime advantage of GD becomes **irrelevant**

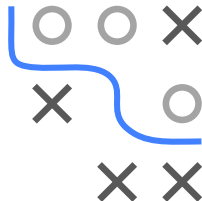
LOGISTIC REGRESSION (CORR.)

- In case of correlated features the results are very similar:



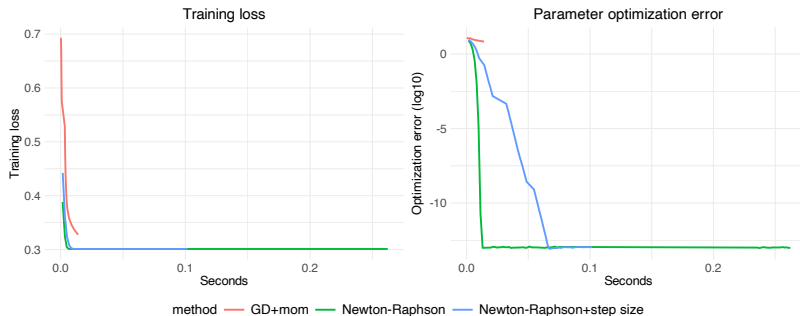
Dotted lines indicate global minimizers.

- **NR's** performance is unaffected by feature correlation
- **GD** iterates become “warped” compared to before



RUNTIME COMPARISON (CORR.)

- Previous conclusions on runtime comparison for independent features carry over to correlated feature case:



Observations:

- **NR** steps are indeed slower than **GD** steps ($\approx 4\times$ here)
- Overall **NR** is strongly superior to **GD** wrt optim error and speed

