MATH 565: Lecture 13 (02/24/2026)

Today:
* saddle points
* convergence problems
* Trust region method

Newton Method: Challenges

1. Ill-conditioned Hessian

Tikhonov regularization term $R = \frac{\lambda}{2}\|\bar{w}\|^2$ $(\lambda > 0)$

$\Rightarrow$ $HR = \lambda I$

(Hessian of R)

Adding λI to HJ (for J without regularizer) will fix these problems.

It also helps when HJ is indefinite. In this case, choose $\lambda > \left| \min_{\lambda_i < 0} \{\lambda_i\} \right|$, where λ_i's are the eigenvalues of HJ (without regularizer).

Adding λI to HJ makes is positive definite (PD).

In practice, one often chooses a small value, e.g., $\lambda = 0.05, 0.01$, etc.

2. Saddle Points (for nonconvex loss functions)

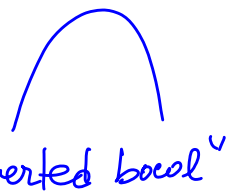
Recall $J(w) = -w^3 + 4w^2 + 1$ from Lecture 11, where

$\nabla J = -3w^2 + 8w$ and

$HJ = -6w + 8$.

We saw the quadratic approximation of J at $w_0 = 2$ as $F(w) = -2w^2 + 12w - 7$, which is concave.

(@ $w_0 = 2$)



"inverted bowl"

The quadratic approximation of $J(w)$ at $w_0' = 1$ is

$$G(w) = 4 + (w-1)5 + \frac{1}{2}(w-1)^2 2$$

@ $w_0' = 1$

$$= w^2 + 3w, \quad \text{which is convex.}$$

This $J(w)$ does not have a saddle point (inflection point).

But $J(w) = -w^3$ has a saddle point at $w_0 = 0$.

↳ $\nabla = 0$, but it is not a local max or min

We saw saddle points in 2D previously, e.g.,

$$E(x,y) = x^2 - y^4, \quad HE = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \text{ at } (0,0).$$

If we consider $J(\bar{w}) = w_1^2 - w_2^2$,

$$HJ = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}, \text{ which also has the classic saddle point at } (0,0).$$

Here, approaching from w_1 or w_2 directions give very different quadratic approximations.

This problem with saddle points for additively separable loss functions can become significant in high dimensions.

In many such cases, gradient descent (variants, e.g., Adam) could avoid saddle points more effectively.

↳ $J(\bar{w}) = \sum_{i=1}^{n} J_i(w_i)$, and if each J_i has k_i saddle points, then J has $\prod_{i=1}^{n} k_i$ saddle points.

3. Convergence Problems

(d=1)

A simple example of logistic regression SVM in 1D for $n=2$.

Let $D = \{\underbrace{(1,1)}_{x_1,y_1}, \underbrace{(2,1)}_{x_2,y_2}\}$ (2 points; trivially well-separated)

$$J(\omega) = \ln(1+e^{-\omega}) + \ln(1+e^{-2\omega}) \quad \left(= \sum_{i=1}^{n} \ln(1+e^{-y_i(\vec{\omega}^T \vec{x}_i)})\right)$$

$J(\omega) \to 0$ as $\omega \to \infty$

$$\Rightarrow \nabla J(\omega) = \frac{-e^{-\omega}}{1+e^{-\omega}} + \frac{-2e^{-2\omega}}{1+e^{-2\omega}}$$

$$= -\left(\frac{1}{1+e^{\omega}} + \frac{2}{1+e^{2\omega}}\right)$$

$$\text{and } HJ(\omega) = \frac{e^{\omega}}{(1+e^{\omega})^2} + \frac{4e^{2\omega}}{(1+e^{2\omega})^2}$$

* "Starting too far" problem

$\omega_0 = -5 \quad \nabla J(\omega_0) \approx -3, \quad HJ(\omega_0) \approx 0.007,$

Newton update: $\omega \leftarrow \omega_0 - \frac{\nabla J}{HJ} \approx -5 + 438 = 433.$

good movement, but no idea of convergence! → has overshot the "good" range

* "The infinite pursuit" problem

$\omega_0 = 10, \quad \nabla J(\omega_0) \approx -4.5 \times 10^{-5}, \quad H \approx 4.5 \times 10^{-5}$

Newton update: $\omega \leftarrow \omega_0 - \frac{\nabla J}{HJ} \approx 11$

↳ the updates will continue as $10 \to 11 \to 12 \to \cdots$

We need an explicit limit on the # iterations for Newton method to stop.

We consider several approaches to address these challenges.

A. Trust Region Method (TRM)

In line search, one chooses a descent direction, and then chooses the step size, which tells how far to move along the descent direction.

TRM: choose step size first, and then choose a good descent direction.

For $J(\bar{w})$, let $F(\bar{w})$ be its second order Taylor approximation at $\bar{w}_t$ (w at iteration t):

$$F(\bar{w}) = J(\bar{w}_t) + (\bar{w}-\bar{w}_t)^T \nabla J(\bar{w}_t) + \frac{1}{2}(\bar{w}-\bar{w}_t)^T H J(\bar{w}_t)(\bar{w}-\bar{w}_t)$$

Then TRM solves

$$\min F(\bar{w})$$
$$\text{s.t. } \|\bar{w}-\bar{w}_t\| \le \delta_t$$

δ_t: trust radius → search only within a "small" radius (in a region that we can trust)

Similar to Newton method, but we do not necessarily go all the way to the "bottom of quadratic bowl".

If $\delta_t << 1$ (small)

TRM $\approx$ gradient descent (and not Newton).

How to choose δ_t? We use the

improvement ratio $I_t = \dfrac{J(\bar{w}_t) - J(\bar{w}_{t+1})}{F(\bar{w}_t) - F(\bar{w}_{t+1})}$.

While TRM optimizes $F(\bar{w})$, our original goal is to minimize $J(\bar{w})$, the input loss function. Hence we compare the decrease, i.e., improvement, in $J(\bar{w})$ to that of $F(\bar{w})$.

Usually, $I_t < 1$.

Choosing $\delta_t \ll 1$ may give $I_t \approx 1$, but progress is slow.

We choose δ_t with I_t as a "hint":

- if $I_t = 0.25$, say, (i.e., small),
  then $\delta_{t+1} = \dfrac{\delta_t}{2}$.
- if $I_t = 0.75$, say, (i.e., large), and
  $\|\bar{w} - \bar{w}_t\| = \delta_t$ at optimum → constraint is tight
  then $\delta_{t+1} = 2\delta_t$
- else, $\delta_{t+1} = \delta_t$ (no change)
- if $I_t < 0$,
  then $\bar{w}_{t+1} = \bar{w}_t$ and (re-)solve
  with $\delta_{t+1} = \dfrac{\delta_t}{2}$.

↓

thus, TRM is using the current trust region is used to the max; hence, we need to increase the trust radius.