COMP5211: Machine Learning
Lecture 3
Logistics

• Form your group
  • Group registration: Due next Monday
    • Submit your team members & project title & project abstract
• Homework 1 will release this week
Optimization

Goal

• Goal: find the minimizer of a function
  • \( \min_w f(w) \)

• For now we assume \( f \) is twice differentiable
Optimization
Convex function

• A function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function

• $\iff$ the function $f$ is below any line segment between two points on $f$:

  • $\forall x_1, x_2, \forall t \in [0,1]$,
  
  • $f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$
Optimization

Convex function

- A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex function

- $\iff$ the function $f$ is below any line segment between two points on $f$:
  - $\forall x_1, x_2, \forall t \in [0,1],$
  - $f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$

- Strictly convex:
  $f(tx_1 + (1 - t)x_2) < tf(x_1) + (1 - t)f(x_2)$
Another equivalent definition for differentiable function:

- $f$ is convex if and only if $f(x) \geq f(x_0) + \nabla f(x_0)^T(x - x_0)$, $\forall x, x_0$
Optimization

Convex function

• Convex function:

  • (For differentiable function) \( \nabla f(w^*) = 0 \iff w^* \) is a global minimum

• If \( f \) is twice differentiable \( \Rightarrow \)

  • \( F \) is convex if and only if \( \nabla^2 f(w) \) is positive semi-definite

  • Example: linear regression, logistic regression, …
Optimization

Convex function

• Strict convex function:
  • $\nabla f(w^*) = 0 \iff w^*$ is the unique global minimum
  • Most algorithms only converge to gradient=0
  • Example: Linear regression when $X^TX$ is invertible
Optimization
Convex vs Nonconvex

- Convex function:
  - $\nabla f(x) = 0 \rightarrow$ Global minimum
  - A function is convex if $\nabla^2 f(x)$ is positive definite
  - Example: linear regression, logistic regression, …
- Non-convex function:
  - $\nabla f(x) = 0 \rightarrow$ Global min, local min, or saddle point
  - Most algorithms only converge to gradient =0
  - Example: neural network, …
Optimization

Gradient descent

• Gradient descent: repeatedly do
  
  • $w^{t+1} \leftarrow w^t - \alpha \nabla f(w^t)$
  
  • $\alpha > 0$ is the step size

• Generate the sequence $w^1, w^2, \ldots$

  • Converge to stationary points ($\lim_{t \to \infty} \|\nabla f(w^t)\| = 0$)
Optimization

Gradient descent

• Gradient descent: repeatedly do
  - $w_{t+1} \leftarrow w_t - \alpha \nabla f(w_t)$
  - $\alpha > 0$ is the step size
• Generate the sequence $w^1, w^2, \ldots$
  - Converge to stationary points
    \[ \lim_{t \to \infty} \| \nabla f(w^t) \| = 0 \]
  - Step size too large $\Rightarrow$ diverge;
  - too small $\Rightarrow$ slow convergence
Optimization
Why gradient descent

• At each iteration, form an approximation function of $f(\cdot)$:

  $$f(w + d) \approx g(d) := f(w^t) + \nabla f(w^t)^{d} + \frac{1}{2\alpha}||d||^2$$

• Update solution by $w^{t+1} \leftarrow w^t + d^*$

  $$d^* = \arg \min_d g(d)$$

  $$\nabla g(d^*) = 0 \Rightarrow \nabla f(w^t) + \frac{1}{\alpha}d^* = 0 \Rightarrow d^* = -\alpha \nabla f(w^t)$$

• $d^*$ will decrease $f(\cdot)$ if $\alpha$ (step size) is sufficiently small
Optimization

Illustration of gradient descent

• Form a quadratic approximation

\[
f(w + d) \approx g(d) := f(w^t) + \nabla f(w^t)d + \frac{1}{2\alpha} \|d\|^2
\]
Optimization
Illustration of gradient descent

\[ g(d) \approx f(w^t + d) \]

- Minimize \( g(d) \)
  
  \[ \nabla g(d^*) = 0 \implies \nabla f(w^t) + \frac{1}{\alpha} d^* = 0 \implies d^* = -\alpha \nabla f(w^t) \]
Optimization

Illustration of gradient descent

- Update $w$
  
  $w^{t+1} = w^t + d^* = w^t - \alpha \nabla f(w^t)$
Optimization

Illustration of gradient descent

- Update \( w \)
  - \( w^{t+1} = w^t + d^* = w^t - \alpha \nabla f(w^t) \)
Optimization
Illustration of gradient descent

\[ g(d) \approx f(w^{t+1} + d) \]
Optimization

Illustration of gradient descent

\[ g(d) \approx f(w^{t+1} + d) \]
Optimization
When will it diverge

Can diverge \((f(w^t) < f(w^{t+1}))\) if \(g\) is not an upper bound of \(f\)

\[ f(w^t) < f(w^{t+1}), \text{ diverge because } g\text{'s curvature is too small } \]
Optimization

When will it converge

Always converge \( (f(w^t) > f(w^{t+1})) \) if \( g \) is an upper bound of \( f \)

\[ f(w^t) > f(w^{t+1}), \text{ converge when } g \text{'s curvature is large enough} \]
Optimization

Convergence

• A differential function $f$ is said to be L-Lipschitz continuous:
  \[ \|f(x_1) - f(x_2)\|_2 \leq L\|x_1 - x_2\|_2 \]

• A differential function $f$ is said to be L-smooth: its gradient are Lipschitz continuous:
  \[ \|\nabla f(x_1) - \nabla f(x_2)\|_2 \leq L\|x_1 - x_2\|_2 \]

• And we could get
  \[ \nabla^2 f(x) \preceq LI \]
  \[ f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}L\|y - x\|^2 \]
Optimization

Convergence

• Let $L$ be a Lipchitz constant ($\nabla^2 f(x) \leq LI$ for all $x$)

  • Theorem: gradient descent converges if $\alpha < \frac{1}{L}$

• In practice, we do not know $L$ ...
  • Need to tune step size when running gradient descent
Optimization
Convergence

- Let $L$ be a Lipchitz constant ($\nabla^2 f(x) \leq LI$ for all $x$)

- Theorem: gradient descent converges if $\alpha < \frac{1}{L}$

- Why?
Optimization
Convergence

• Let $L$ be a Lipchitz constant \( (\nabla^2 f(x) \leq LI \text{ for all } x) \)

• Theorem: gradient descent converges if $\alpha < \frac{1}{L}$

• Why?
  
  • When $\alpha < 1/L$, for any $d$,
    
    \[
g(d) = f(w^t) + \nabla f(w^t)^T d + \frac{1}{2\alpha} \|d\|^2
    \]

    \[
    > f(w^t) + \nabla f(w^t)^T d + \frac{L}{2} \|d\|^2
    \]

    \[
    \geq f(w^t + d)
    \]

  • So, $f(w^t + d^*) < g(d^*) \leq g(0) = f(w^t)$

  • In formal proof, need to show $f(w^t + d^*)$ is sufficiently smaller than $f(w^t)$
Optimization
Gradient descent convergence rate

• Suppose $f$ is convex and differentiable and its gradient is Lipschitz continuous, then if we run gradient for $t$ iterations with a fixed step $\alpha \leq \frac{1}{L}$, it will yield a solution that satisfies:

  $f(w^t) - f(w^*) \leq \frac{\|w^0 - w^*\|_2^2}{2\alpha t}$

• Proof
Optimization
Convergence

• Let $L$ be a Lipchitz constant $(\nabla^2 f(x) \leq LI$ for all $x$)

  • Theorem: gradient descent converges if $\alpha < \frac{1}{L}$

• In practice, we do not know $L$ ...
  • Need to tune step size when running gradient descent
Optimization
Applying to logistic regression

gradient descent for logistic regression

- Initialize the weights $w_0$
- For $t = 1, 2, \cdots$
  - Compute the gradient

$$\nabla f(w) = -\frac{1}{N} \sum_{n=1}^{N} \frac{y_n x_n}{1 + e^{y_n w^T x_n}}$$

- Update the weights: $w \leftarrow w - \eta \nabla f(w)$
- Return the final weights $w$
Optimization
Applying to logistic regression

• When to stop?
  • Fixed number of iterations, or
  • Stop when $\|\nabla f(w)\| < \epsilon$

gradient descent for logistic regression

- Initialize the weights $w_0$
- For $t = 1, 2, \cdots$
  - Compute the gradient
    $$\nabla f(w) = -\frac{1}{N} \sum_{n=1}^{N} \frac{y_n x_n}{1 + e^{y_n w^T x_n}}$$
  - Update the weights: $w \leftarrow w - \eta \nabla f(w)$
- Return the final weights $w$
Optimization
Line search

• In practice, we do not know $L$ ...
  • Need to tune step size when running gradient descent
• Line Search: Select step size automatically (for gradient descent)
Optimization
Line search

• The back-tracking line search:
  • Start from some large $\alpha_0$
  • Try $\alpha = \alpha_0, \alpha_0/2, \alpha_0/4, \ldots$
    • Stop when $\alpha$ satisfies some sufficient decrease condition
Optimization
Line search

• The back-tracking line search:
  • Start from some large $\alpha_0$
  • Try $\alpha = \alpha_0, \alpha_0/2, \alpha_0/4, \ldots$
    • Stop when $\alpha$ satisfies some sufficient decrease condition
  • A simple condition: $f(w + \alpha d) < f(w)$
Optimization

Line search

• The back-tracking line search:
  • Start from some large $\alpha_0$
  • Try $\alpha = \alpha_0, \alpha_0/2, \alpha_0/4, \ldots$
    • Stop when $\alpha$ satisfies some sufficient decrease condition
  • A simple condition: $f(w + \alpha d) < f(w)$
    • Often works in practice but doesn’t work in theory
Optimization
Line search

• The back-tracking line search:
  • Start from some large $\alpha_0$
  • Try $\alpha = \alpha_0, \alpha_0/2, \alpha_0/4, \ldots$
    • Stop when $\alpha$ satisfies some sufficient decrease condition
• A simple condition: $f(w + \alpha d) < f(w)$
  • Often works in practice but doesn’t work in theory
• A (provable) sufficient decrease condition $f(w + \alpha d) \leq f(w) + c_1 \alpha \nabla f(w)^T d$ (armijo condition)
• $\nabla f(w + \alpha d)^T d \geq c_2 \nabla f(w)^T d$ (curvature)
  • + armijo = wolfe condition
• For constant $c_1, c_2 \in (0,1)$
Optimization

Line search

gradient descent with backtracking line search

- Initialize the weights $w_0$
- For $t = 1, 2, \cdots$
  - Compute the gradient
    
    $$d = -\nabla f(w)$$
  - For $\alpha = \alpha_0, \alpha_0/2, \alpha_0/4, \cdots$
    - Break if $f(w + \alpha d) \leq f(w) + \sigma \alpha \nabla f(w)^T d$
  - Update $w \leftarrow w + \alpha d$
- Return the final solution $w$