Gradient Descent

Machine Learning – CSE546

Kevin Jamieson
University of Washington

April 24, 2019

Machine Learning Problems

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters: $$\sum_{i=1}^n \ell_i(w)$$

Each $\ell_i(w)$ is convex.

Sum of convex functions is convex

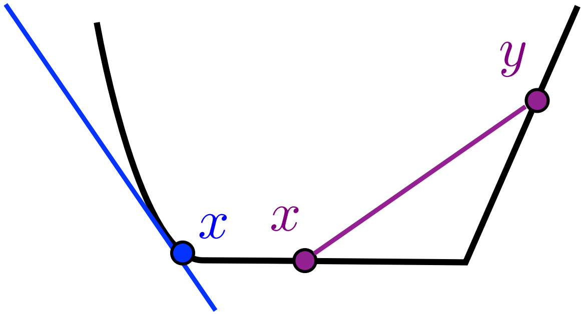
Machine Learning Problems

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters: $$\sum_{i=1}^n \ell_i(w)$$

Each $\ell_i(w)$ is convex.



g is a subgradient at x if
$$f(y) \geq f(x) + g^T(y-x)$$

f convex:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) \qquad \forall x, y, \lambda \in [0,1]$$
$$f(y) \geq f(x) + \nabla f(x)^T(y-x) \qquad \forall x, y$$

$$f((1-\lambda)x+\lambda y) \le (1-\lambda)f(x)+\lambda f(y) \iff f(y)-f(x) \ge f'(y)(y-x)$$

$$f\big((1-\lambda)x+\lambda y\big) \le (1-\lambda)f(x)+\lambda f(y)$$

$$= f(x)+\lambda\big(f(y)-f(x)\big)$$

Divide λ by both sides

$$f(y)-f(x) \ge \frac{f(x+\lambda(y-x))-f(x)}{\lambda}$$

$$= \frac{f(x+\lambda(y-x))-f(x)}{\lambda(y-x)}(y-x)$$

$$\xrightarrow{\lim \lambda\to 0} = f'(x)(y-x)$$

Machine Learning Problems

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters:

$$\sum_{i=1}^n \ell_i(w)$$

Each $\ell_i(w)$ is convex.

Logistic Loss: $\ell_i(w) = \log(1 + \exp(-y_i\, x_i^T w))$

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

Least squares

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters:

$$\sum_{i=1}^n \ell_i(w)$$

Each $\ell_i(w)$ is convex.

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

How does software solve: $\frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

Least squares

- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters: $\sum_{i=1}^{n} \ell_i(w)$

Each $\ell_i(w)$ is convex.

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

How does software solve: $\frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

...its complicated:
(LAPACK, BLAS, MKL…)

Do you need high precision?
Is X column/row sparse?
Is $\widehat{w}_{LS}$ sparse?
Is $\mathrm{X}^T\mathrm{X}$ "well-conditioned"?
Can $\mathrm{X}^T\mathrm{X}$ fit in cache/memory?

Taylor Series Approximation

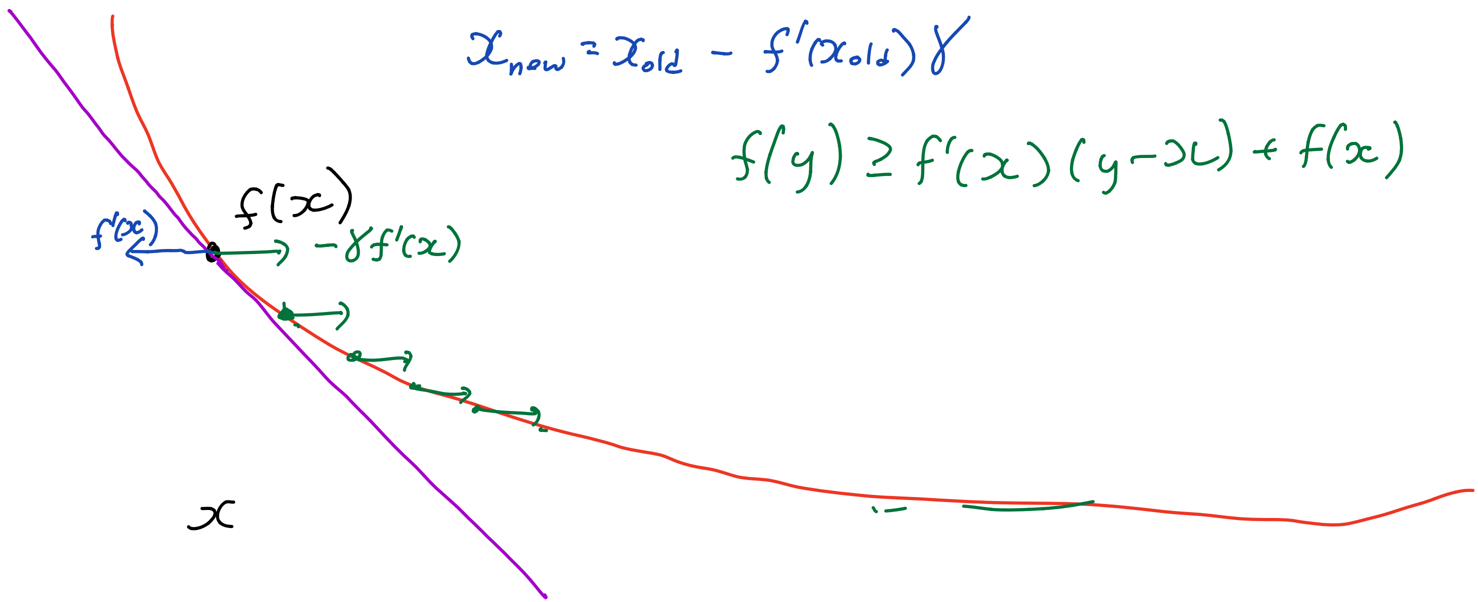
- Taylor series in one dimension:

$$\underbrace{f(x+\delta) = f(x) + f'(x)\delta}_{} + \tfrac{1}{2}f''(x)\delta^2 + \ldots$$

- Gradient descent:

$x_{new} = x_{old} - f'(x_{old})\gamma$

$f(y) \geq f'(x)(y-x) + f(x)$



Taylor Series Approximation

- Taylor series in **d** dimensions:

$$f(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v + \ldots$$

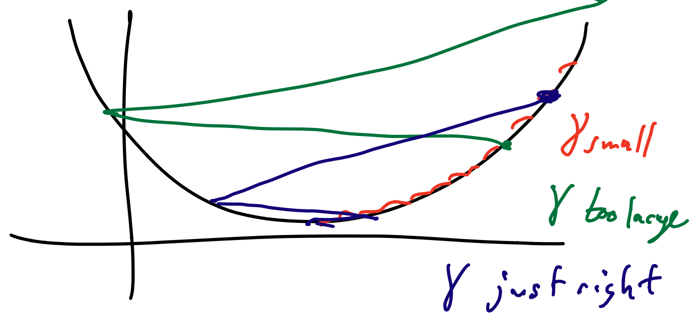
- Gradient descent:

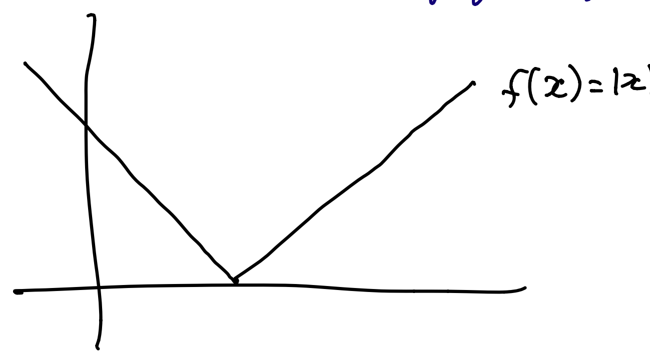
$x_0 = 0, \quad t = 0$

while $\|\nabla f(x_t)\|_2^2 > \varepsilon \;\Big|\; |f(x_{t+1}) - f(x_t)| > \varepsilon$

$x_{t+1} = x_t - \gamma \nabla f(x_t)$

$t = t+1$





Gradient Descent $f(w) = \frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

$$\nabla f(w) =$$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

$$w_* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

$$w_{t+1} - w_* =$$

Gradient Descent

$$f(w) = \frac{1}{2}||Xw - y||_2^2$$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

$$\nabla f(w) = \mathbf{X}^T(\mathbf{X}w - \mathbf{y}) = \mathbf{X}^T\mathbf{X}w - \mathbf{X}^T\mathbf{y}$$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

$$= (I - \eta \mathbf{X}^T\mathbf{X})w_t + \eta \mathbf{X}^T\mathbf{y}$$

$$w_* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

$$(w_{t+1} - w_*) = (I - \eta \mathbf{X}^T\mathbf{X})(w_t - w_*) \underbrace{- \eta \mathbf{X}^T\mathbf{X}w_* + \eta \mathbf{X}^T\mathbf{y}}_{=0}$$

Gradient Descent $f(w) = \frac{1}{2}||Xw - y||_2^2$

$$w_{t+1} = w_t - \eta \nabla f(w_t)$$

$$(w_{t+1} - w_*) = (I - \eta X^T X)(w_t - w_*) = (I - \eta X^T X)(I - \eta X^T X)(w_{t-1} - w_*)$$

$$= (I - \eta X^T X)^{t+1}(w_0 - w_*)$$

Example: $X = \begin{bmatrix} 10^{-3} & 0 \\ 0 & 1 \end{bmatrix}$ $y = \begin{bmatrix} 10^{-3} \\ 1 \end{bmatrix}$ $w_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ $w_* = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \eta \begin{bmatrix} 10^{-6} & 0 \\ 0 & 1 \end{bmatrix} \right)^{t+1} (w_0 - w_*)$$

$$(w_{t+1} - w_*)[0] = (1 - \eta 10^{-6})^{t+1} (w_0 - w_*)[0]$$

$$\text{''} \quad \text{''} \quad [1] = (1 - \eta)^{t+1} \quad \text{''} \quad \text{''} \quad [1]$$

Taylor Series Approximation

- Taylor series in one dimension:

$$f(x+\delta) = f(x) + f'(x)\delta + \frac{1}{2}f''(x)\delta^2 + \ldots$$

- Newton's method:

Taylor Series Approximation

- Taylor series in **d** dimensions:

$$f(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v + \ldots$$

- Newton's method:

Newton's Method $f(w) = \frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

$$\nabla f(w) =$$

$$\nabla^2 f(w) =$$

$$v_t \text{ is solution to} : \nabla^2 f(w_t) v_t = -\nabla f(w_t)$$

$$w_{t+1} = w_t + \eta v_t$$

Newton's Method $f(w) = \frac{1}{2}||\mathrm{X}w - \mathrm{y}||_2^2$

$$\nabla f(w) = \mathrm{X}^T(\mathrm{X}w - \mathrm{y})$$

$$\nabla^2 f(w) = \mathrm{X}^T\mathrm{X}$$

$$v_t \text{ is solution to} : \nabla^2 f(w_t) v_t = -\nabla f(w_t)$$

$$w_{t+1} = w_t + \eta v_t$$

For quadratics, Newton's method converges in one step! (Not a surprise, why?)

$$w_1 = w_0 - \eta(\mathrm{X}^T\mathrm{X})^{-1}\mathrm{X}^T(\mathrm{X}w_0 - y) = w_*$$

General case

In general for Newton's method to achieve $f(w_t) - f(w_*) \leq \epsilon$:

So why are ML problems overwhelmingly solved by gradient methods?

Hint: $v_t \text{ is solution to} : \nabla^2 f(w_t) v_t = -\nabla f(w_t)$

General Convex case $f(w_t) - f(w_*) \leq \epsilon$

Clean converge nice proofs: Bubeck

Newton's method:

$$t \approx \log(\log(1/\epsilon))$$

Gradient descent:

- f is *smooth* and *strongly convex:* $aI \preceq \nabla^2 f(w) \preceq bI$
- f is *smooth*: $\nabla^2 f(w) \preceq bI$
- f is potentially non-differentiable: $||\nabla f(w)||_2 \leq c$

Nocedal +Wright, Bubeck

Other: BFGS, Heavy-ball, BCD, SVRG, ADAM, Adagrad,…

Revisiting… Logistic Regression

Machine Learning – CSE546

Kevin Jamieson
University of Washington

April 24, 2019

Loss function: Conditional Likelihood

- Have a bunch of iid data of the form: $\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i | x_i, w) \qquad P(Y = y | x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

$$= \sigma(y w^T x)$$

$$f(w) = \arg\min_w \sum_{i=1}^n \log(1 + \exp(-y_i\, x_i^T w))$$

$$\nabla f(w) = \sum_{i=1}^n \nabla \log(1 + \exp(-y_i x_i^T w))$$

$$= \sum_{i=1}^n \frac{1}{1 + \exp(-y_i x_i^T w)} \cdot \exp(-y_i x_i^T w)(-y_i x_i)$$