Algorithms for optimizing convex functions

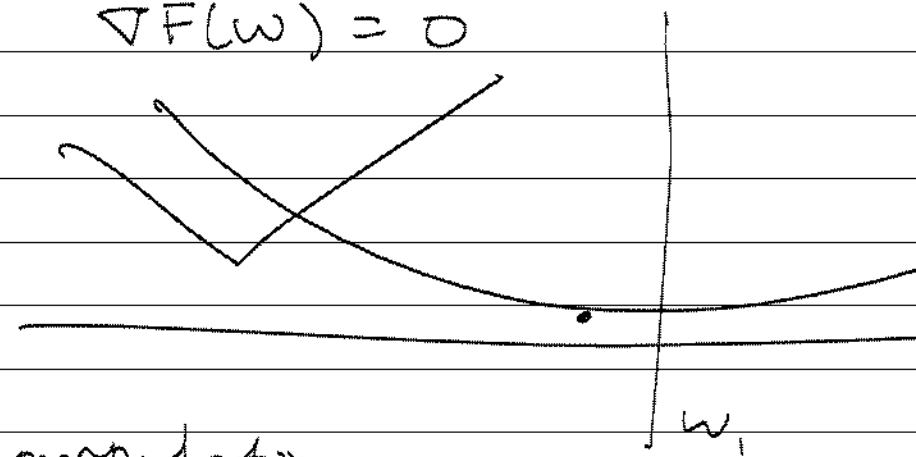
Note Title 13-06-2010

$$\min_{w} F(w)$$

Assume: not easy to solve for $\nabla F(w) = 0$

$w \in \mathbb{R}$ $d = 1$

$F(w)$ is expensive to evaluate.



I) Zero-th order algorithms
- do not require gradient computation
- applicable to non-differential functions

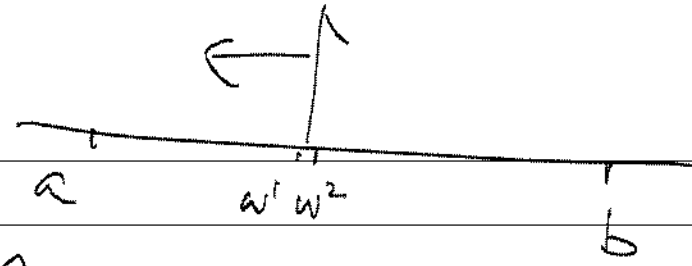
II First-order algorithms

- require gradient & function computation.
- applicable to function which are not twice differentiable.

III Second-order algorithms
- require Hessian

IV Semi-Second order algorithms.
- approximate the Hessian

Assume: $w_1 \in [a \ldots b]$



Dichotomous Binary Search Algorithm

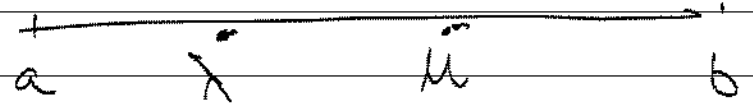
while $|b-a| \geq \varepsilon$

* Pick 2 points w^1 & w^2 δ-apart in the middle of a & b
* if $F(w^1) > F(w^2)$ then $a = w^1$
  else $b = w^2$

$(0.5)^{T/2} \simeq \varepsilon$

$$T = 2\left(\frac{\log \varepsilon}{\log 0.5}\right)$$

Golden section Search Algorithm



$\lambda = a + (1-\alpha)(b-a)$ where $\alpha = 0.618$

$\mu = a + \alpha(b-a)$

calculate: $f(\lambda)$ & $f(\mu)$

while $(|a-b| > \varepsilon)$ {

if $f(\lambda) > f(\mu)$ {

$a = \lambda$

$\lambda = \mu$; $f(\lambda) = f(\mu)$

$\mu = a + \alpha(b-a)$

calculate $f(\mu)$

} else {

$b = w$

$w = \lambda \quad f(w) = f(\lambda)$

$\lambda = a + (1-\alpha)(b-a)$

calculate $f(\lambda)$

}

$(0.618)^{T-1} = \varepsilon$

First order methods

Gradient of the Function is avaible.

Modified bisection algorithm

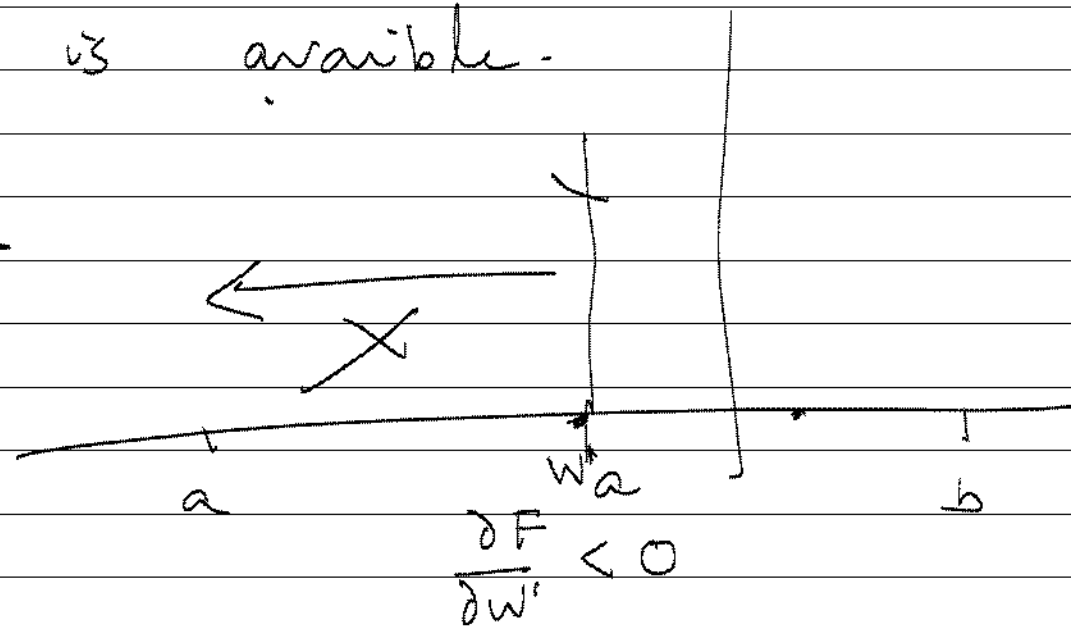
$(0.5)^T \leq \varepsilon$

while $|a - b| \geq \varepsilon$ {

$w' = \frac{a+b}{2}$

$dw' = \frac{\partial F}{\partial w'}$

if $(dw' < 0)$

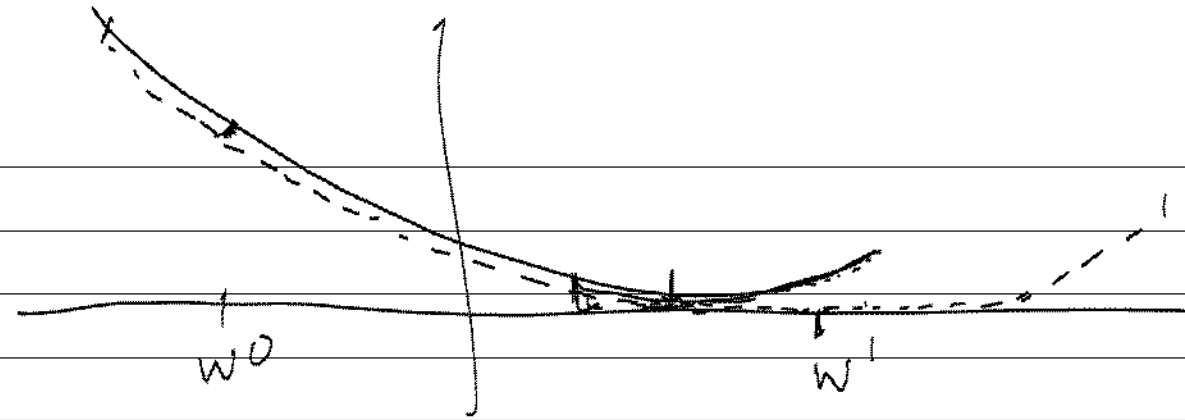


$a = w^1$

else

$b = w^1$

}



Second-order methods

$F(w)$ is twice differentiable

$$\tilde{F}(w) = F(w^0) + \nabla F(w^0)^T (w - w^0) + \frac{1}{2} (w - w^0)^T H_F(w^0) (w - w^0)$$

$\tilde{F}(w)$ is convex.

$$\min_w \tilde{F}(w) \Leftrightarrow \nabla \tilde{F}(w) = 0$$

1d

$$\tilde{F}(w) = F(w^0) + \frac{\partial}{\partial w} F(w^0)(w - w^0) + \frac{1}{2} (w - w^0)^2 \frac{\partial^2 F(w^0)}{\partial w^2}$$

$$\nabla \tilde{F}(w) = 0 + \frac{\partial F(w^0)}{\partial w} + (w - w^0)\frac{\partial^2 F(w^0)}{\partial w^2} = 0$$

$$w = -\frac{1}{\frac{\partial^2 F(w^0)}{\partial w^2}}\left(\frac{\partial F(w^0)}{\partial w}\right) + w^0 \qquad (1-d)$$

$$-H_F(w^0)^{-1}\nabla F(w^0) + w^0 \qquad (n-d)$$

$w^0 = 0$

while ($\|\nabla F(w^0)\| \geq \epsilon$) {

$w^{t+1} = w^t - H_F(w^t)^{-1}\nabla F(w^t)$

$t = t + 1$

}

Warning: may not converge

Optimizing multidimensional function

Descent-based algorithms $w \in R^d$

$\tilde{F}(w) = F(w^0) + \nabla F(w^0)^T \cdot (w - w^0)$

$w = \lambda \vec{e} + w^0$ $\vec{e} \equiv$ unit vector in d-dimensions

$\lambda \in R$ $\lambda \geq 0$

$\tilde{F}(w^0 + \lambda \vec{e}) = F(w^0) + \lambda \nabla F(w^0)^T \vec{e}$

≤ 0

$t = 0$

w^0 = arbitrary initial value

while $\left(|\nabla F(w^t)^0| \geq \varepsilon \right)$ {

Find a descent direction $\vec{e}$ $\nabla F(w^t) \cdot \vec{e} \leq 0$

$\lambda^* = \arg\min F(w^t + \lambda \vec{e})$ X

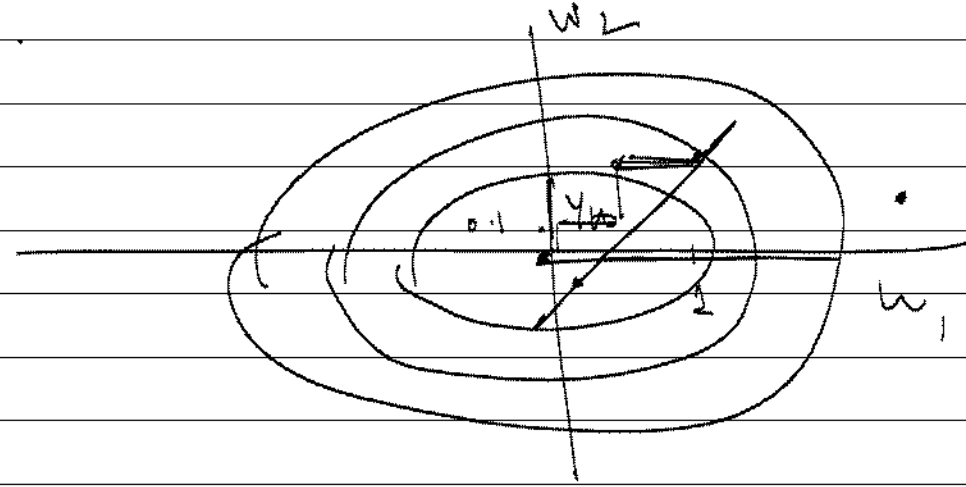
$w^{t+1} = w^t + \lambda^* \vec{e}$

$t = t+1$

}

1-d convex minimiza problem.

HW To prove that $G(\lambda) = F(w^0 + \lambda \vec{e})$ is convex in λ if $F(w)$ is convex in w.



$d=2$

$$f(w_1, w_2) = \frac{1}{2}(w_1^2 + 10 w_2^2)$$

$$w^0 = (10, 1)^T$$

$$\vec{e} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

$$g(\lambda) = f(w^0 + \lambda \vec{e})$$

$$= \frac{1}{2}\left(\left(w_1^0 + \frac{\lambda}{\sqrt{2}}\right)^2 + 10\left(w_2^0 + \frac{\lambda}{\sqrt{2}}\right)^2\right)$$

$$g(\lambda) = \frac{1}{2}\left(\left(10 + \frac{\lambda}{\sqrt{2}}\right)^2 + 10\left(1 + \frac{\lambda}{\sqrt{2}}\right)^2\right)$$

$$\min_{\lambda} g(\lambda) \equiv \frac{\partial g}{\partial \lambda} = 0$$

$$g'(\lambda) = \left(10 + \frac{\lambda}{\sqrt{2}}\right)\frac{1}{\sqrt{2}} + 10\left(1 + \frac{\lambda}{\sqrt{2}}\right)\frac{1}{\sqrt{2}} = 0$$

$$\Rightarrow \quad 20 + \frac{11\lambda}{\sqrt{2}}$$

$$\Rightarrow \quad \lambda^* = -\frac{20\sqrt{2}}{11}$$

$$w^1 = w^0 + \lambda^* \vec{e}$$

$$= \begin{pmatrix} 10 \\ 1 \end{pmatrix} - \frac{20\sqrt{2}}{11} \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 10 \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{20}{11} \\ \frac{20}{11} \end{pmatrix}$$

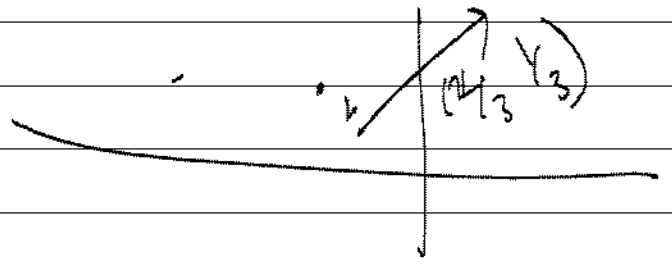
$$F(w^0) = 55$$

$$F(w^1) = 36.\sim$$

Gradient descent algorithm:

$$\vec{e} = -\nabla F(w^t)$$

$$\| \nabla F^T(w^t) \cdot \vec{e} \|$$



Simpler optimization algorithm

avoids $\lambda^* = \underset{\lambda}{\operatorname{argmin}} F(w^t + \lambda \vec{e})$

pick λ_t = small value.

$$w^{t+1} = w^t - \lambda_t \nabla F(w^t)$$

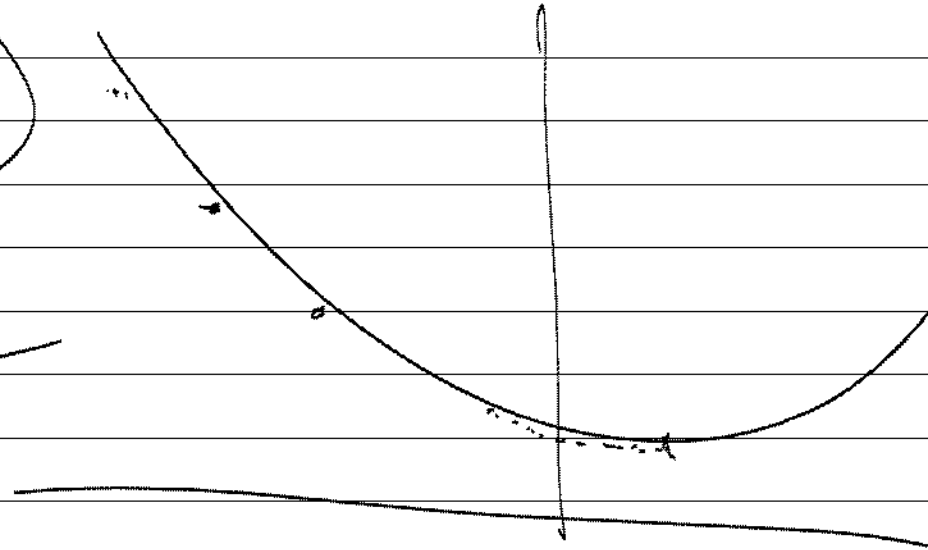
$$\tilde{F}(w^{t+1}) = F(w^t) - \lambda \underbrace{\nabla \tilde{F}(w^t) \cdot \nabla F(w^t)}_{\geq 0}$$

λ is small enough ≥ 0

$$\tilde{F}(w^{t+1}) \approx F(w^{t+1})$$

$$F(w^{t+1}) \leq F(w^t)$$

$$\lambda_t = \frac{\text{Constant}}{t}$$



Co-ordinate descent algorithm

$\vec{e} = [1\ 0\ 0\ 0 \cdots 0]$

or $[0\ 1\ 0 \cdots 0]$

$\vdots$

$[0 \cdots 0\ 1]$

binary unit vector.

change w^t in only one dimension to get a w^{t+1}

Second-order algorithm.

$$w^{t+1} = w^t - \underbrace{H(w^t)^{-1} \nabla F(w^t)}_{\vec{e}}$$

may not converge.

Instead pick $w^{t+1} = \underset{\lambda \geq 0}{\operatorname{argmin}} F\left(w^t - \lambda H(w^t)^{-1} \nabla F(w^t)\right)$

Gives optimum in one step for quadratic

Very efficient in terms convergence rate

but requires Hessian computation.

$d \times d$

Approximate 2nd order method via

* pseudo - second order methods

* conjugate gradient methods.

Applying for logistic loss minimization.

$$\min_w F(w) = \min_w \sum_{i=1}^{N} \log\left(1 + e^{-w x^i y_i}\right)$$

$$\nabla F(w) = \sum_{i=1}^{N} \frac{e^{-w x^i y_i}}{1 + e^{-w x^i y_i}} \left(-x^i y_i\right)$$

$w = 0 ; t = 0$

while $(\|\nabla F(w)\| \geq \epsilon)$ {

/* calculate gradient at w^t */

$g \leftarrow 0$

for $i = 1$ to N ← expensive.

$$\nabla F_i(w) = \frac{e^{-w^t \cdot x^i y_i}}{1 + e^{-w^t \cdot x^i y_i}} (-x^i \cdot y_i)$$

$O(d)$

$g = g + \nabla F_i(w)$ ←

$\lambda^t = \min_{\lambda} F(w^t + \lambda g)$ ← line search

optional.

$w^{t+1} = w^t - \lambda^t g$

$\alpha^t = \frac{\text{constant}}{t}$;

$t = t + 1$ }

Off-the shelf optimization algorithm : eg: LBFGS

the loop above implemented by the algorithm

the ML trainer only implements code for evaluating $F(w)$ & $\nabla F(w)$ at any given w.