CS 5/4780

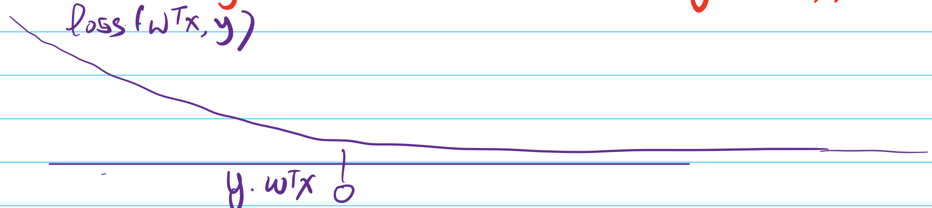
Gradient Descent and Beyond

Logistic regression recap: directly model $P(Y=y \mid X=x) = \dfrac{1}{1+\exp(-y w^{T} x)}$

$$\hat{w}_{MLE} = \underset{w}{\operatorname{argmax}} \prod_{i=1}^{n} P_w(Y = y_i, X = x_i)$$

$$= \underset{w}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{n} \overbrace{\log(1+\exp(-y_i w^T x_i))}^{\text{logistic loss}}}_{\ell(w)}$$

Logistic loss: $\text{loss}(w^T x, y) = \log(1+\exp(-y \cdot w^T x))$



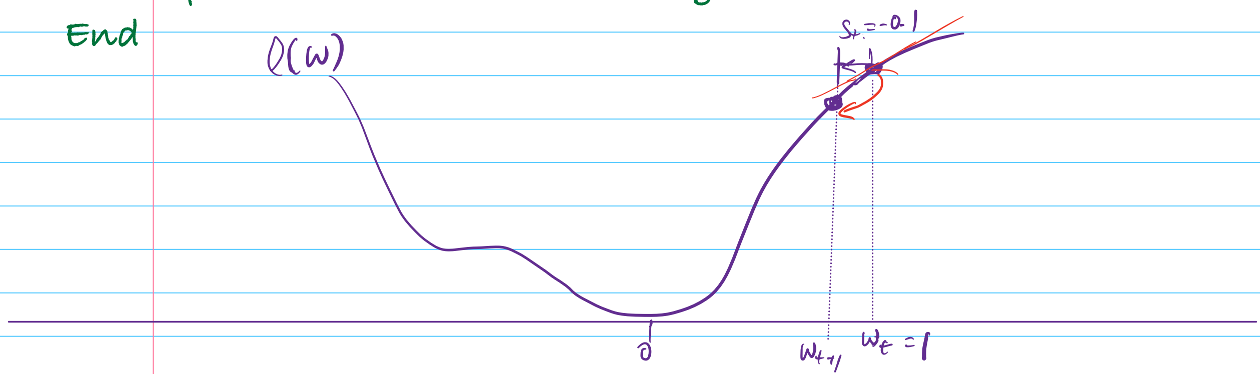
1. Unfortunately above problems don't have closed form solution
2. Majority of ML algorithms solve problem of form $\underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \ \ell(w)$
3. Assume the objective is differentiable (twice differentiable)

General Scheme: Input initial guess w_0, $t = 0$

While not converged:

1. Pick direction $s_t \in \mathbb{R}^d$, $t = t+1$
2. $w_{t+1} = w_t + s_t$
3. If $\|w_t - w_{t+1}\| < \delta$ Then converged

End



which direction to choose?

Taylor's theorem: $f(x) = f(0) + x f'(0) + \frac{x^2}{2!} f''(0) + \ldots + \frac{x^k}{k!} f^{(k)}(0) + \ldots$

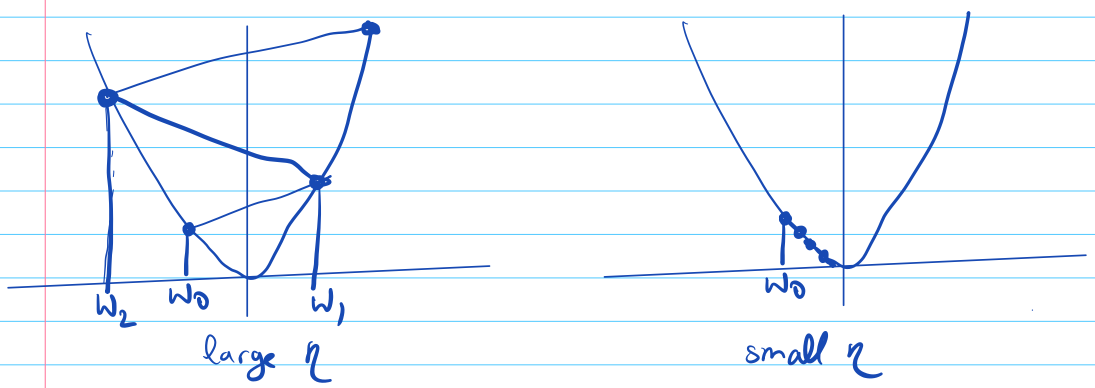
1st order: $\ell(w+s) \approx \ell(w) + s^T \nabla \ell(w) + O(\|s\|^2)$ (when $\|s\|$ is small)

Where $\nabla \ell(w)[i] = \frac{\partial \ell(w)}{\partial w[i]}$

Let $s = -\eta \nabla \ell(w)$, for small enough η.

$$\underbrace{\ell(w - \eta \nabla \ell(w)) \approx \ell(w) - \eta \|\nabla \ell(w)\|^2 \le \ell(w)}_{\text{progress}}$$

1. η too small will make progress slow
2. η too large can even diverge



1. We can use step size η_t that varies with iteration t
2. When loss is convex (Eg. Logistic loss) $\eta_t = \frac{1}{\sqrt{t}}$ works in theory and we are guaranteed convergence to minima
3. When gradients are on average small we might want step size to be large and vice versa

Demo

A problematic example for gradient descent:

$$\ell(w) = w[1]^2 + 0.01\, w[2]^2$$

Demo

1. if we set η too large, we diverge on first coordinate
2. if we set η small, first coordinate is good but we are very slow on second coordinate

AdaGrad: Adaptive Gradient Descent (different stepsize for different coordinates)

Input initial guess w_0, $z_0 = 0$, $t = 0$
While not converged:
1. $g_t = \nabla \ell(w_t)$, $t = t+1$
2. $\forall i \in \{1,\ldots,d\}$, $z_t[i] = z_{t-1}[i] + g_t[i]^2$
3. $\forall i \in \{1,\ldots,d\}$, $w_{t+1}[i] = w_t[i] - \eta\, g_t[i] / \sqrt{(z_t[i] + \varepsilon)}$
3. If $\|w_t - w_{t+1}\| < \delta$ Then converged

End

Back to $\ell(w) = w[1]^2 + 0.01\, w[2]^2$

Step size for coordinate 2 is a factor of 0.01 times that of coordinate 1

2nd order: $\ell(w+s) \approx \ell(w) + s^T \nabla \ell(w) + \frac{1}{2} s^T \nabla^2 \ell(w)\, s + O(\|s\|^3)$

Where $\nabla^2 \ell(w)$ is the hessian matrix of ℓ at w. $\nabla^2 \ell(w)[i,j] = \dfrac{\partial^2 \ell(w)}{\partial w[i]\, \partial w[j]}$

Newton's method: Find s that Minimizes above second order approximation $s = -(\nabla^2 \ell(w))^{-1} \nabla \ell(w)$. $w_{t+1} = w_t - (\nabla^2 \ell(w_t))^{-1} \nabla \ell(w_t)$

1. Typically second order approximation is appropriate near minima, so warm start with gradient descent and finish with Newton's method
2. Hessian size d^2 matrix and we need to invert hessian, so expensive in large dimensions

Stochastic Gradient Descent (SGD):

Typically $\ell(w) = \frac{1}{n} \sum_{i=1}^{n} loss(w; x_i, y_i)$

In this case instead of computing $\nabla \ell(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla loss(w; x_i, y_i)$ on each iteration, we can instead sub-sample a set B_t consisting of b samples of D on each iteration and approximate gradient by

$$\hat{g}_t(w) = \frac{1}{b} \sum_{(x,y) \in B_t} \nabla loss(w; x, y)$$

$b = 1$ or constant often suffices