Lecture 2

Optimization Methods(continued)

Convergence guarantees for GD

$$
\begin{aligned}
&start\ with\ some\ w^{(0)}\\
&For\ t=0\ to\ T:\\
&\quad w^{(t+1)}=w^{(t)}-\eta\nabla F(w^{(t)})\\
&\quad t=t+1
\end{aligned}
$$

Many results for GD (and many variants) on convex objectives.

They tell you how many iterations t (in terms of ε) are needed to achieve

$$F(w^{(t)})-F(w^*)\leqslant\varepsilon$$

Even for non-convex objectives, some guarantees exist:

e.g. how many iterations t (in terms of ε) are needed to achieve

$$||\nabla F(w^{(t)})||\leqslant\varepsilon$$

that is, how close is $w^{(t)}$ as an approximate stationary point

for convex objectives, stationary point $\Rightarrow$ global minimizer.

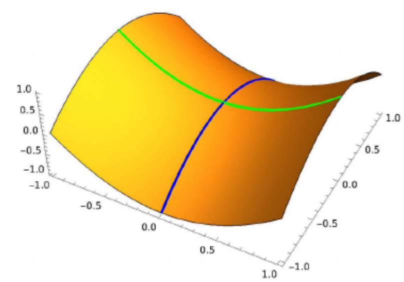
for non-convex objectives, what does it mean?

Stationary points: non-convex objectives

It can be a local minimizer or even a local/global maximizer. (but the latter is not an issue for GD)

It can also be neither a local minimizer nor a local maximizer

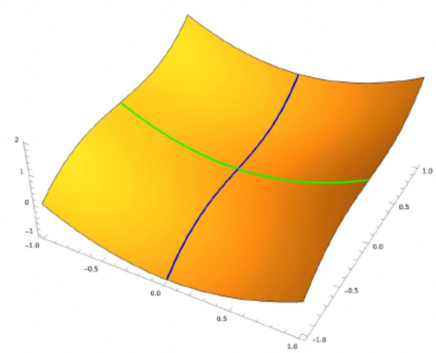
eg. $f(w) = w_1^2 - w_2^2,\ \nabla f(w) = (2w_1, -2w_2),\ (0,0)$ point is stationary. It's a local max for direction $w_2\ (w_1 = 0)$, but a local min for direction $w_1\ (w_2 = 0)$.



Point like $(0,0)$ is known as a saddle point.

But not all saddle look like 'saddle':

$f(w) = w_1^2 - w_2^3,\ \nabla f(w) = (2w_1, 3w_2^2),\ (0,0)$ is stationary but not local min/max for direction w_2 when $w_1 = 0$.



In this case, GD gets stuck at $(0,0)$ for any initial point with $w_2 \geqslant 0$ and small η

Even worse, distinguishing local min and sanddle point is generally NP-hard.

Stochastic Gradient Descent

SGD: keep moving in the noisy negative gradient direction

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \tilde{\nabla} F(w^{(t)})$$

where $\tilde{\nabla} F(w^{(t)})$ is a random variable(called stochastic gradient) s.t.

$$\mathbb{E}[\tilde{\nabla} F(w^{(t)})] = \nabla F(w^{(t)}) \quad (unbiasedness)$$

Key point: it could be much faster to obtain a stochastic gradient!

Similar convergence guarantees, usually needs more iterations but each iteration takes less time.

- GD/SGD coverages to a stationary point.
- for convex objectives, this is all we need.
- for nonconvex objectives, can get stuck at local minimizers or "bad" saddle points (random initialization escapes "good" saddle points)
- recent research shows that many problems have no "bad" saddle points or even "bad" local minimizers.
- justify the practical effectiveness of GD/SGD (default method to try)

Second-order methods (Newton)

GD: first-orders Taylor approximation

$$F(w) \approx F(w^{(t)}) + \nabla F(w^{(t)})^T (w \cdot w^{(t)})$$
$$f(y) \approx f(x) + f'(x)(y-x) + \frac{f''(x)}{2}(y-x)^2$$

what about a second-order Taylor approximation?

$$F(w) \approx F(w^{(t)}) + \nabla F(w^{(t)})^T (w - w^{(t)}) + \frac{1}{2}(w - w^{(t)})^T H_t (w - w^{(t)})$$
$$where\ H_t = \nabla^2 F(w^{(t)}) \in \mathbb{R}^{(d)}\ is\ Hessian\ of\ F\ at\ w^{(t)}$$
$$H_{i,j} = \frac{\partial^2 F(w)}{\partial w_i \partial w_j}|_{w=w^{(t)}}$$
$$Def: \tilde{F}(w) = 2nd\ order\ approximation$$

$\nabla \tilde{F}(w) = 0, \ \therefore$

$$\nabla F(w^{(t)}) + H_t w - \frac{H_t}{2} w^{(t)} - \frac{1}{2} H_t w^{(t)} = 0$$
$$H_t w = H_t w^{(t)} - \nabla F(w^{(t)})$$
$$w = w^{(t)} - H_t^{-1} \nabla F(w^{(t)})$$

Newton method: $w^{(t+1)} \leftarrow w^{(t)} - H_t^{-1} \nabla F(w^{(t)})$

GD: $w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla F(w^{(t)})$

Newton's Method	**GD**
no learning rate	need to tune η
super fast convergence	slower convergence
Know & invert Hessian (inversion needs $O(d^3)$ time naively)	fast ($O(d)$ time)

Linear Classifiers

input: $x \in \mathbb{R}^d$

output: $y \in [C] = \{1, 2, \cdots, C\}$

goal: learn a mapping $f : \mathbb{R}^d \to [C]$

Number of classes: $C = 2$

Labels: $\{+1, -1\}$

$$sign(w^T x) = \begin{cases} +1 \; if \; w^T x > 0 \\ -1 \; if \; w^T x \leqslant 0 \end{cases}$$

Def: the function class of separating hypo-planes

$$\mathcal{F} = \{f(x) = sign(w^T x) : w \in \mathbb{R}^d\}$$

it still makes sense for "almost" linearly separable data

Most common loss:

$$l(f(x), y) = \mathbb{1}(f(x) \neq y)$$

Loss as a function of $yw^T x$

$$l_{0-1}(yw^T x) = \mathbb{1}(yw^T x \leqslant 0)$$

$0 - 1$ loss is not convex, and is NP-hard in general.

perceptron loss: $l(z) = \max\{0, -z\}$.

Use a convex surrogate loss:

hinge loss: $l(z) = \max\{0, 1 - z\}$

logistic loss: $l(z) = \log(1 + \exp(-z))$

Find ERM:

$$w^* = \arg\min_{w \in \mathbb{R}^d} \frac{1}{n}(\sum_{i=1}^{n} l(y_i w^T x_i))$$
$$where \; l(\cdot) \; is \; a \; convex \; surrogate \; loss$$

Perceptron

$0-1$ **Loss and SGD**

$$\begin{aligned} F(w) &= \frac{1}{n}\sum_{i=1}^{m} l(y_i w^T x_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \max\{0, -y_i w^T x_i\} \end{aligned}$$

Let's try GD|SGD.

$$gradient: \begin{cases} 0, z \geqslant 0 \\ -1, z \leqslant 0 \end{cases}$$

Gradient is

$$\nabla F(w) = \frac{1}{n}\sum_{i=1}^{n} -\mathbb{1}[y_i w^T x_i \leqslant 0] y_i x_i$$

only misclassified examples count

$$GD: w \leftarrow w + \frac{\eta}{n}\sum_{i=1}^{n} \mathbb{1}[y_i w^T x_i \leqslant 0] y_i x_i$$

need the entire training set for every GD update.

How to get a stochastic gradient?

pick one example $i \in [n]$ uniformly at random, let

$$\nabla \tilde{F}(w^{(t)}) = -\mathbb{1}[y_i w^T x_i \leqslant 0] y_i x_i$$

Unbiased, why?

$$\begin{aligned} \mathbb{E}[\tilde{\nabla} F(w^{(t)})] &= \frac{1}{n}\sum_{i=1}^{n} -\mathbb{1}[y_i w^T x_i \leqslant 0] y_i x_i \\ &= \nabla F(w^{(t)}) \end{aligned}$$

SGD update: $w \leftarrow w + \eta\, \mathbb{1}(y_i w^T x_i \leqslant 0) y_i x_i$

This is fast! one data-point per update

objective function of most ML tasks is a finite sum. trick applies generally.

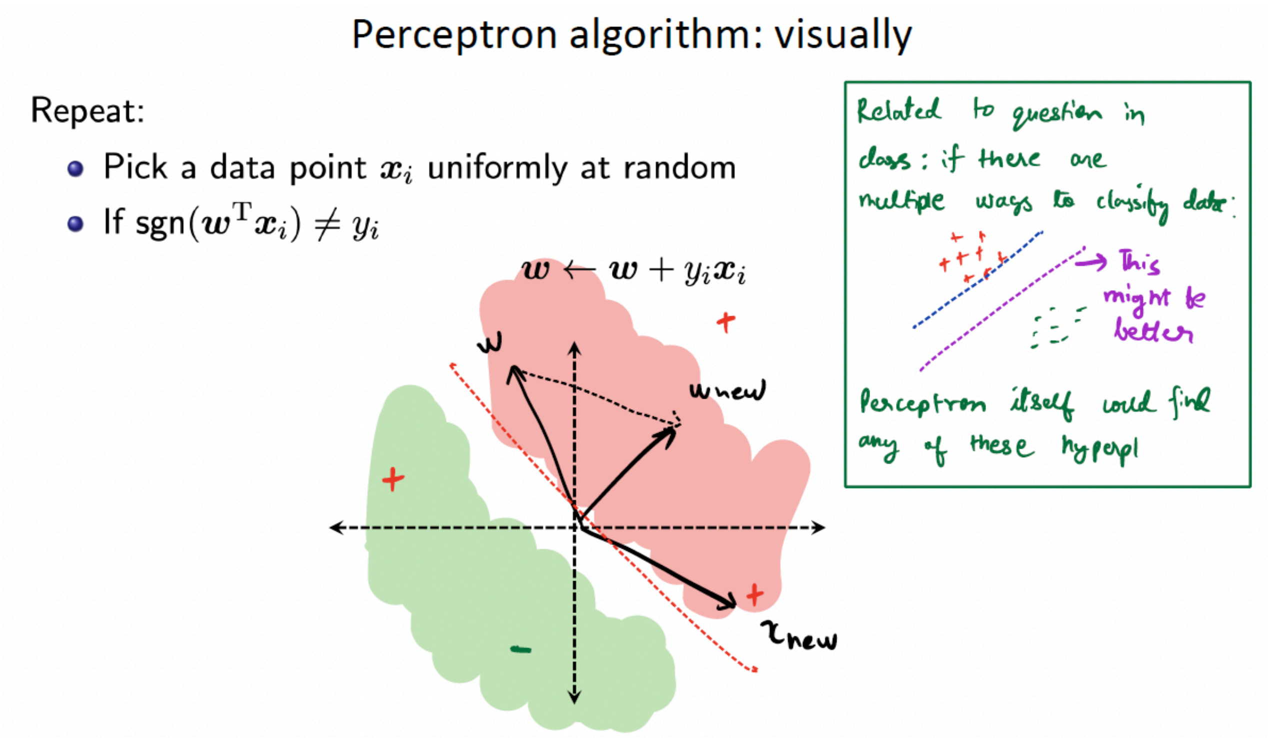
Algorithm

SGD with $\eta = 1$ on perceptron loss:

$$
\begin{aligned}
&initialize\ w = 0\\
&Repeat\\
&\quad pick\ x_i \sim Unif(x_1, \cdots, x_n)\\
&\quad If\ sign(w^T x_i) \neq y_i:\\
&\qquad w \leftarrow w + y_i x_i
\end{aligned}
$$

Intuition: say that w makes mistake on (x_i, y_i)

$$
\begin{gathered}
y_i w^T x_i < 0\\
consider\ w' = w + y_i x_i\\
y_i (w')^T x_i = y_i w^T x_i + y_i^2 x_i^T x_i\\
if\ x_i \neq 0: y_i (w')^T x_i > y_i w^T x_i
\end{gathered}
$$



If training set is linearly separable: Perceptron converges in a finite number of steps; Training error is 0 .

There are also guarantees when the data are not linearly separable

Logistic Regression

$$
\begin{aligned}
F(w) &= \frac{1}{n} \sum l(y_i w^T x_i)\\
&= \frac{1}{n} \sum \frac{1}{1 + \exp(-y_i w^T x_i)}
\end{aligned}
$$

Instead of $\{\pm 1\}$, predict the probability (regression on probability)

Sigmoid function

sigmoid + linear model:

$$\mathbb{P}(y \pm 1|\mathcal{X}, w) = \sigma(w^T x)$$
$$where\ \sigma(z) = \frac{1}{1+e^{-z}}\ (sigmoid)$$

$\sigma(z) = \frac{1}{1+e^{-z}}$: between 0 and 1 (good as probability)

$\sigma(w^T x) \geq 0.5 \Leftrightarrow w^T x \geq 0$, consistent with predicting the label with $sign(w^T x)$

larger $w^T x \Rightarrow$ larger $\sigma(w^T x) \Rightarrow$ higher confidence in label 1

$\sigma(z) + \sigma(-z) = 1$ for all z

The probability of label -1 is:

$$P(y = -1|x; w) = 1 - P(y = +1|x; w)$$
$$= 1 - \sigma(w^T x) = \sigma(-w^T x)$$

Therefore, we can model $P(y|x; w) = \sigma(y w^T x) = \frac{1}{1+e^{-y w^T x}}$

MLE(Maximum likelihood estimation)

Specifically, the probability of seeing labels $y_1, \cdots, y_n$ given $x_1, \cdots, x_n$ as a function of some w is:

$$P(w) = \prod_{i=1}^{N} P(y_i|x_i; w)$$

find w^* that maximizes the probability $P(w)$:

$$\begin{aligned}
w^* &= \arg\max_w P(w) \\
&= \arg\max_w \sum_{i=1}^{n} \ln P(y_i|x_i; w) \\
&= \arg\min_w \sum_{i=1}^{n} -\ln P(y_i|x_i; w) \\
&= \arg\min_w \sum_{i=1}^{n} \ln(1 + e^{-y_i w^T x_i}) \\
&= \arg\min_w \sum_{i=1}^{n} l(y_i w^T x_i) \\
&= \arg\min_w F(w)
\end{aligned}$$

Minimizing logistic loss is exactly doing MLE for the sigmoid model!

SGD to logistic loss:

$$
\begin{aligned}
w &\leftarrow w - \eta \nabla_w l(y_i w^T x_i) \\
&= w - \eta (\frac{-e^{-z}}{1+e^{-z}}|_{z=y_i w^T x_i}) y_i x_i \\
&= w + \eta \sigma(-y_i w^T x_i) y_i x_i \\
&= w + \eta \mathbb{P}(-y_i | x_i; w) y_i x_i
\end{aligned}
$$

This is a soft version of perceptron

$$\mathbb{P}(-y_i|x_i;w) \; versus \; \mathbb{1}[y_i \neq sign(w^T x_i)]$$

Unbiased, why?

$$\mathbb{E}[\tilde{\nabla} F(w)] = \nabla F(w) \; (i \; is \; drawn \; uniformly \; from \; [n])$$

$$Chain \; Rule: \frac{\partial(\log(1+e^{-z}))}{\partial z} = \frac{-e^{-z}}{1+e^{-z}}$$

$$\sigma(-z) = 1 - \sigma(z) = 1 - \frac{1}{1+e^{-z}} = \frac{e^{-z}}{1+e^{-z}}$$