Into the World of Influence

Prof. Nick Vincent
Simon Fraser University

2024-01-31

Training Data Influence

Our starting point

- Hammoudeh and Lowd's 2023 Survey
- H&L for short in this slide deck
- Covers 'early' stats work (1980s), more 'recent' ML work (2017)
- Gives us a taxonomy of influence approaches and definitions

Basic definition

Imagine we have ourselves a nice, trained ‘final model’. How much of it’s ‘value’ boils to down to row 537 from our training set?

The grand question of influence

- How much did playing high school sports impact your ability to be successful in CS courses?
- How much did that extra serving of fries contribute to my stomach ache last night?
- How much did this particular set of lectures slides contribute to your learning outcomes in this course?

In general, these are all very hard questions. Quantiative social scientists, especially in economics and sociology, like to try to answer them when possible using experiments and clever analysis approaches.

But often in real life, we'll just never know how your CS career would have gone if you {did / did not} play high school {basketball / hockey / baselball}.

But models *should* be traceable

- In the case of models, however… why can't we figured out exactly how much row 537 contributed. It's all just math, right?

Why we want to do this

- a number of reasons given from H&L
- detect anamolies
- distribution shifts
- measurement error
- human labeling errors
- adversarial actors

Influence and a data economy?

- another big one: potential implications for a new post-AI ‘data economy’

Can we calculate influences?

- Depends on the definition of influence we use and the particular case
- definitions can differ quite a bit
- sometimes ‘provably hard’[1]
- we can always estimate (but how good are the estimates)

Terms

- Let's call our models M
- Let's call our training data D
- Let's call our row of interest D_i (for now, more notation coming)

Different ways to relate M to D

- influence analysis, aka data valuation, aka data attribution: which pieces of training data get credit (and how much) for a specific model output
- Leave-one-out influence = different between M's' performance with D_i vs. without D_i

Brute force influence

“We can easily get all influences for all rows in D. Just retrain n times!”

Well, if retraining was free…

Influence estimation

- Research area has sprung up around estimating training data influence[1]

About H&L's paper

- It's a survey of the many perspective, definitions, and estimation approaches for training data influence
- The authors taxonomize the approaches

For the purposes of our course

- You do not need to understand everything in this paper for CMPT 419! But we will be able to get a lot out of it.

H&L's notation

$[r]$ is set of integers $1, \ldots, r$, i.e. the integers from 1 to r.

$A \overset{m}{\sim} B$ means that A is a set of cardinality m (it has m items) drawn uniformly at random from some other set B.

2^A is the *power set* of set A (set of all subsets, all combinations big and small).[1]

$A \setminus B$ is set subtraction (remove stuff in set B from set A)[2]

More notation

We use bold-face 1 as the indicator function for some condition a.

$\mathbb{1}[a]$. The indicator function is equal to 1 when the predicate a is true (i.e. a is the output of some boolean check).

More notation

$x \in \mathscr{X} \subseteq \mathbb{R}^d$

is a feature vector

$y \in \mathscr{Y}$ is our target

D in our training set

it's a set of n tuples z_i, each tuple is (x_i, y_i)

- subscript i means it's a train examples
- subscript te means it's a test examples, e.g. z_{te} is (x_{te}, y_{te})

Our model is f

It's a function that maps from to

Paramaters are $\theta \in \mathbb{R}^p$

$p := |\theta|$, i.e. p is our number of parameters.

We have some loss function l

We have the empirical risk for an instance z,
$L(z;\theta) := l(f(x;\theta), y)$.

For loss and risk, smaller is better.

Some assumptions (and more notation)

We also assume $p >> d$ (many more parameters than columns in our data)

Using first order optimization (e.g. gradient descent) with T iterations

Start with initial params $\theta^{(0)}$ and we get new params $\theta^{(t)}$ at each time step t

Other hyperparams:

- learning rate $\eta^{(t)} > 0$
- weight decay λ

We won't worry to much about hyperparameters when we're dealing with data valuation – we kind of assume "all that stuff is sorted, let's worry about the data!"

Gradients

We get *training gradients* defined as

$\nabla_\theta L(z_i; \theta^{(t)})$

That is, the gradient (with respect to parameters theta) of the empirical risk of instance z_i for parameters at time step t.[1]

This will be important!

Hessian

empirical risk hessian is

$H_\theta^{(t)} := \frac{1}{n} \sum_{z_i \in D} \nabla_\theta^2 L(z_i; \theta^{(t)})$

Data missing some subset

$D \setminus z_i$ is D without instance z_i

We can write $\theta^{(}_{D \setminus z_i} t)$ to mean the parameters when trained with that instance missing

You might begin to imagine why this will be useful!

Some vocab

- *proponents*, *excitatory examples*: training examples with positive influence, loss goes down when the example is added, which is “good”.
- *opponents*, *inhibitory examples*: training examples with negative influence, loss goes up when the example is added, which is “bad”.
- *pointwise influence*: effect of single instance on single metric (e.g. test loss)