Learning a logistic regression from Aggregated Data

Alexandre Gilotte

David Rohde

Click prediction models and Privacy sandbox

Partner	Publisher	Ad Size	More features…	Click ?
42	A	Small	…	0
43	A	Large	…	0
42	B	Large	…	1
...	…	…		0

Learning model of $P(Y=1|X=x)$

$$Loss := \Sigma_i \; Loss(f(x_i, y_i))$$

- Eg logistic regression
- Gradient descent, …

Privacy sandbox:
- Dataset no longer available!
- Instead, “Aggregated data”

Aggregated data ?

Learning P(Y | X) from these tables ??

Tables counting displays and clicks

- On subsets of variables
- Tables may be overlapping
- Also, noise may be added to get differential privacy guarantees.

Partner	Publisher	Nb Displays	Nb Clicks
42	A	10000	600
42	B	55000	1000
43	A	20000	500
43	B	8000	300
…	…	…	…

Partner	Ad size	Nb Displays	Nb Clicks
42	Small	100000	5500
…	…	…	…

Publisher	Ad size	Nb Displays	Nb Clicks
A	Small	30000	700
…	…	…	…

Formalizing the aggregated data

- Unobserved dataset (x_i, y_i) iid
- Quadratic kernel $\mathcal{K}$ mapping x to $\{0;1\}^D$

Observed aggregated data:

$$d := \sum_i \mathcal{K}(x_i)$$

$$c := \sum_i y_i \cdot \mathcal{K}(x_i)$$

Partner = 42,
Publisher = B,
Size = Small

x

$\xrightarrow{\mathcal{K}}$

0	Partner 42 and pub A
1	Partner 42 and pub B
…	…
0	Partner 43 and pub A
…	…
1	Partner 42 and size small
…	…
1	Publisher B and size small
0	Publisher B and size large
…	…

$\mathcal{K}(x) \in \{0;1\}^D$

Proposed approach

Modeling
Choose a parametric model on the **joined** distribution of features X and labels Y

$$\mathbb{P}_\theta(X = x, Y = y)$$

Training
Select theta maximizing the likelihood of observed event:

$$Argmax_\theta \ \mathbb{P}_\theta(D = d, C = c)$$

Intractable for most models ?!

Aggregated data are a realization of the random variables:

$$D := \sum_i \mathcal{K}(X_i)$$

$$C := \sum_i Y_i \cdot \mathcal{K}(X_i)$$

Predicting

With the conditional law :

$$\mathbb{P}_\theta(Y = 1 | X = x) = \frac{\mathbb{P}_\theta(Y = 1, X = x)}{\mathbb{P}_\theta(Y = 1, X = x) + \mathbb{P}_\theta(Y = 0, X = x)}$$

Loglinear model

Modeling

Parametric model loglinear in $\mathcal{K}(X)$:

$$\mathbb{P}_{\mu,\theta}(X = x, Y = y) := \frac{\exp\left(\mathcal{K}(x) \cdot \mu + y \cdot \mathcal{K}(x) \cdot \theta\right)}{Z_{\mu,\theta}}$$

- « Random Markov Field »

Predicting

$$\mathbb{P}_{\mu,\theta}(Y = y | X = x) = \sigma\left(\mathcal{K}(x) \cdot \theta\right)$$

- No Z, and no µ
- Looks like a logistic regression with kernel $\mathcal{K}$

Normalization constant. Intractable!?

Training

Gradient of the log-likelihood

$$\nabla_\mu \, Log \, \mathbb{P}_{\mu,\theta}(D = d, C = c) = d - \mathbb{E}_{\mu,\theta}(D)$$

$$\nabla_\theta \, Log \, \mathbb{P}_{\mu,\theta}(D = d, C = c) = c - \mathbb{E}_{\mu,\theta}(C)$$

- Exponential family
- Aggregated data are sufficient statistics!

Aggregated data

- Depends only on the model (no data)
- Estimated by Monte Carlo on Gibbs samples.

Experimental Results

On a medium size public Criteo dataset

- Public Criteo dataset
- Quite « small »
  - 11 features
  - 16M examples, 33% clicks

Model	NLLH	Training time
Logistic, 2 order kernel, full dataset	0.091	2h
Markov Random Field (ours)	0.089	120h
Logistic, no kernel, full dataset	0.076	0.2h

- Not far from skyline !

On Criteo Adkdd challenge

- Larger dataset
- 18 features
- ~100M aggregated samples

Model	Logloss Improvement vs Naive
Logistic, 2 order kernel, full dataset	0.311
Challenge winners	0.29 ??
Logistic, no kernel, full dataset	0.289
Markov Random Field (ours)	0.265
Logistic, 2 order kernel, small trainset	0.238

- Still quite far from logistic with full data
- But using *only* aggregated data

Limitations and next steps

Optimization is difficult!

- Gibbs sampling: unefficient with strongly correlated features

Validating **with only aggregated data?**

- Choice of parameters and monitoring model quality by cross validation … on granular (ie non aggregated) data.
- How to avoid this?

Modelling error on P(X)

- Compared to logistic regression, we have to model P(X) instead of using train samples
- With many correlated features, our model on P(X) may be very wrong, leading to worse P(Y|X)
- Higher order aggregation tables ?

Also in the paper

L2 Regularization

- Strong regularization on θ
- Low regularization on μ

Monte Carlo on Gibbs samples

- Marginalize on Y to lower the noise on the gradients of θ

Re-using Gibbs samples between gradient steps

- « Persistent contrastive divergence »

Modelling also the noise

- If aggregtaed data are noisy
- Argmax(P(D+Noise = d, C+Noise =c))

Thank you!