Probit Regression for Large Data Sets via Coresets

Alexander Munteanu Simon Omlor Christian Peters

TU Dortmund University, Germany

December 18, 2021

Probit Regression

Probit Regression

Data:

- n observations: $x_1, \ldots, x_n \in \mathbb{R}^d$
- with n labels: $y_1, \ldots, y_n \in \{-1, 1\}$

Probit Regression

Data:

- n observations: $x_1, \ldots, x_n \in \mathbb{R}^d$
- with n labels: $y_1, \ldots, y_n \in \{-1, 1\}$

Model:

- $y_1, \ldots, y_n$ are realizations of $Y_1, \ldots, Y_n$
- $Y_i \sim Bernoulli(\pi_i), \quad \pi_i = \Phi(x_i^T \beta)$
- $\Phi(\cdot)$ is cdf of standard normal distribution

Probit Regression

Data:

- n observations: $x_1, \ldots, x_n \in \mathbb{R}^d$
- with n labels: $y_1, \ldots, y_n \in \{-1, 1\}$

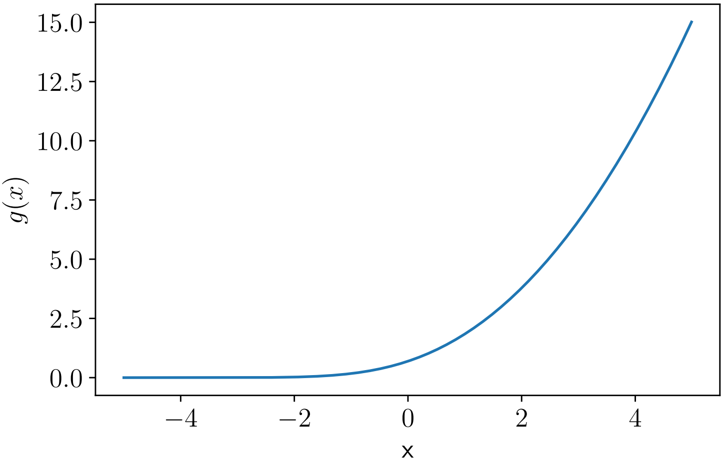
Model:

- $y_1, \ldots, y_n$ are realizations of $Y_1, \ldots, Y_n$
- $Y_i \sim Bernoulli(\pi_i), \quad \pi_i = \Phi(x_i^T \beta)$
- $\Phi(\cdot)$ is cdf of standard normal distribution

Loss function (negative log-likelihood):

$$f(\beta) = \sum_{i=1}^{n} \ln \left(\frac{1}{\Phi(y_i x_i^T \beta)} \right)$$

The Probit Loss



$$g(x) = \ln\left(\frac{1}{\Phi(-x)}\right)$$

The Problem

The Problem

Problems with large datasets:

- Data doesn't fit into main memory
- Limited access to the data (e.g. data streams)

The Problem

Problems with large datasets:

- Data doesn't fit into main memory
- Limited access to the data (e.g. data streams)

Scenarios where this can happen:[1]

- Sensor data from mobile devices, cameras, ...
- Internet logs
- Financial data

[1] see e.g. [Feldman et al., 2020]

The Problem

Problems with large datasets:

- Data doesn't fit into main memory
- Limited access to the data (e.g. data streams)

Scenarios where this can happen:[1]

- Sensor data from mobile devices, cameras, ...
- Internet logs
- Financial data

Conventional optimization algorithms become inefficient!

[1] see e.g. [Feldman et al., 2020]

Our Solution

Our Solution

Select only a small subset (coreset) of the data!
$\Rightarrow$ Fit the model on the coreset.

Our Solution

Select only a small subset (coreset) of the data!
$\Rightarrow$ Fit the model on the coreset.

Challenges:

- Results on coreset must be close to results on original data
  - $\Rightarrow$ Need theoretical guarantees!
- Coreset must be significantly smaller than original data
  - $\Rightarrow$ Otherwise useless!

Our Solution

Select only a small subset (coreset) of the data!
$\Rightarrow$ Fit the model on the coreset.

Challenges:

- Results on coreset must be close to results on original data
  - $\Rightarrow$ Need theoretical guarantees!
- Coreset must be significantly smaller than original data
  - $\Rightarrow$ Otherwise useless!

Main Goal: Develop efficient coreset construction algorithms!

The coresets we want...

The coresets we want...

(1) ...approximate the data well:

- Let $f(\beta)$ be the original loss and $\tilde{f}(\beta)$ be the loss on the coreset
- Then for $\epsilon > 0$ and for all $\beta \in \mathbb{R}^d$ we want:

$$(1-\epsilon)f(\beta) \leq \tilde{f}(\beta) \leq (1+\epsilon)f(\beta)$$

- This criterion will guarantee our approximation quality!

The coresets we want...

(1) ...approximate the data well:

- Let $f(\beta)$ be the original loss and $\tilde{f}(\beta)$ be the loss on the coreset
- Then for $\epsilon > 0$ and for all $\beta \in \mathbb{R}^d$ we want:

$$(1-\epsilon)f(\beta) \leq \tilde{f}(\beta) \leq (1+\epsilon)f(\beta)$$

- This criterion will guarantee our approximation quality!

(2) ...are significantly smaller than our data:

- Coreset sizes logarithmic in n would be a great success
- Given a dataset with $n = 1,000,000,000$ observations, $\log(n) \leq 21$
- Even better: Independent of n!

Our first obstacle

Our first obstacle

Not every data set allows for small coresets.

- Shown in [Munteanu et al., 2018] for logistic regression, but proof is similar for probit regression

Our first obstacle

Not every data set allows for small coresets.

- Shown in [Munteanu et al., 2018] for logistic regression, but proof is similar for probit regression

$\Rightarrow$ **Need to restrict the class of data sets under study!**

- Slightly adapt the concept of μ-complexity from [Munteanu et al., 2018] to probit regression

μ-Complexity

μ-Complexity

μ is a useful parameter introduced by [Munteanu et al., 2018]:

$$\mu = \sup_{\beta \in \mathbb{R}^d \setminus \{0\}} \frac{\sum_{y_i x_i^T \beta > 0} (x_i^T \beta)^2}{\sum_{y_i x_i^T \beta < 0} (x_i^T \beta)^2}$$

μ-Complexity

μ is a useful parameter introduced by [Munteanu et al., 2018]:

$$\mu = \sup_{\beta \in \mathbb{R}^d \setminus \{0\}} \frac{\sum_{y_i x_i^T \beta > 0} (x_i^T \beta)^2}{\sum_{y_i x_i^T \beta < 0} (x_i^T \beta)^2}$$

Finite μ, i.e. μ-complexity ensures:

- That the data is not linearly separable
- That the optimum of the loss function exists and is unqiue

μ-Complexity

μ is a useful parameter introduced by [Munteanu et al., 2018]:

$$\mu = \sup_{\beta \in \mathbb{R}^d \setminus \{0\}} \frac{\sum_{y_i x_i^T \beta > 0} (x_i^T \beta)^2}{\sum_{y_i x_i^T \beta < 0} (x_i^T \beta)^2}$$

Finite μ, i.e. μ-complexity ensures:

- That the data is not linearly separable
- That the optimum of the loss function exists and is unqiue

We show how μ can be used to find small coresets!

How to tackle coreset construction?

How to tackle coreset construction?

Idea: Use random sampling of observations!

How to tackle coreset construction?

Idea: Use random sampling of observations!

Problem: Which sampling distribution to use?

- Uniform equal probability sampling is not a good idea
- It is known to fail when the data has outliers

⇒ We need to give "important" observations a higher priority

How to tackle coreset construction?

Idea: Use random sampling of observations!

Problem: Which sampling distribution to use?

- Uniform equal probability sampling is not a good idea
- It is known to fail when the data has outliers

$\Rightarrow$ We need to give "important" observations a higher priority

Our approach: Use importance sampling based on the sensitivity[2] of each observation!

[2] see [Langberg and Schulman, 2010]

The Sensitivity Framework

The Sensitivity Framework

- Well known algorithmic framework for coreset construction via importance sampling introduced by [Feldman and Langberg, 2011].
- The sensitivity is the worst-case importance of an observation
- Sampling proportionally to upper bounds on the sensitivities can lead to small coresets

$\Rightarrow$ Need to derive tight upper bounds!

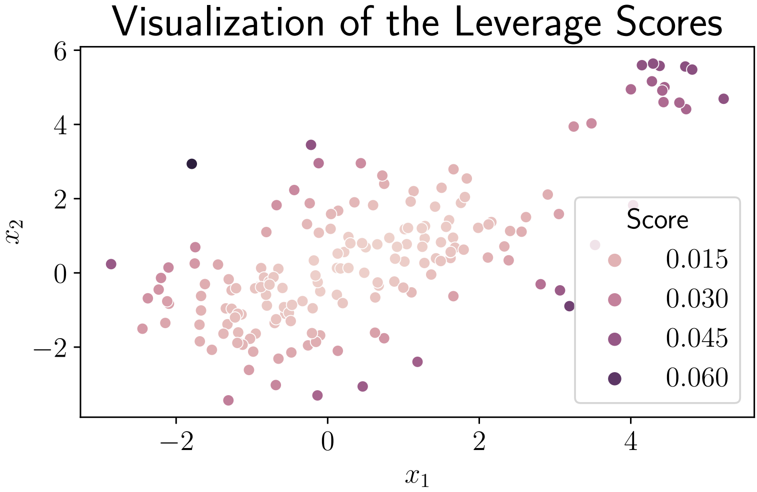
The Sensitivity Framework

- Well known algorithmic framework for coreset construction via importance sampling introduced by [Feldman and Langberg, 2011].
- The sensitivity is the worst-case importance of an observation
- Sampling proportionally to upper bounds on the sensitivities can lead to small coresets

$\Rightarrow$ Need to derive tight upper bounds!

We show that the statistical leverage scores can be used to bound the sensitivities of high-loss points!

Statistical Leverage Scores



First Algorithm: Fast Leverage Score Sampling

First Algorithm: Fast Leverage Score Sampling

First pass: Approximate the leverage scores

- Use sketching techniques by [Clarkson and Woodruff, 2017] to approximate the leverage scores

First Algorithm: Fast Leverage Score Sampling

First pass: Approximate the leverage scores

- Use sketching techniques by [Clarkson and Woodruff, 2017] to approximate the leverage scores

Second pass: Draw the random sample

- Use a reservoir sampler, e.g. by [Chao, 1982]

First Algorithm: Fast Leverage Score Sampling

First pass: Approximate the leverage scores

- Use sketching techniques by [Clarkson and Woodruff, 2017] to approximate the leverage scores

Second pass: Draw the random sample

- Use a reservoir sampler, e.g. by [Chao, 1982]

$$\boxed{\textbf{Coreset size: } O\left(\frac{\mu d^2}{\epsilon^2}\right)\textbf{, independent of } n\textbf{!}}$$

Second Algorithm: Online Leverage Score Sampling

Second Algorithm: Online Leverage Score Sampling

Problem: First algorithm needs two passes.

Second Algorithm: Online Leverage Score Sampling

Problem: First algorithm needs two passes.

Solution: Online approximation of the leverage scores

- Use approximation techniques by [Chhaya et al., 2020]
- Increases coreset size by a factor of $\log(\sigma_{max})$
  - σ_{max} is largest singular value of the data matrix
- Needs $O(d^2)$ update time

Second Algorithm: Online Leverage Score Sampling

Problem: First algorithm needs two passes.

Solution: Online approximation of the leverage scores

- Use approximation techniques by [Chhaya et al., 2020]
- Increases coreset size by a factor of $\log(\sigma_{max})$
  - σ_{max} is largest singular value of the data matrix
- Needs $O(d^2)$ update time

Requires only one pass over the data set!

Experiments

Experiments

Goal: Compare our algorithms to uniform random sampling

- Show that coreset construction is worth the effort

Experiments

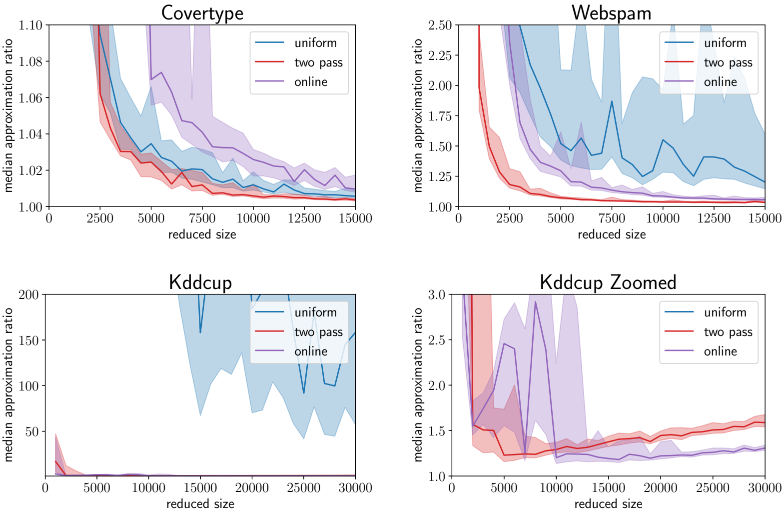
Goal: Compare our algorithms to uniform random sampling

- Show that coreset construction is worth the effort

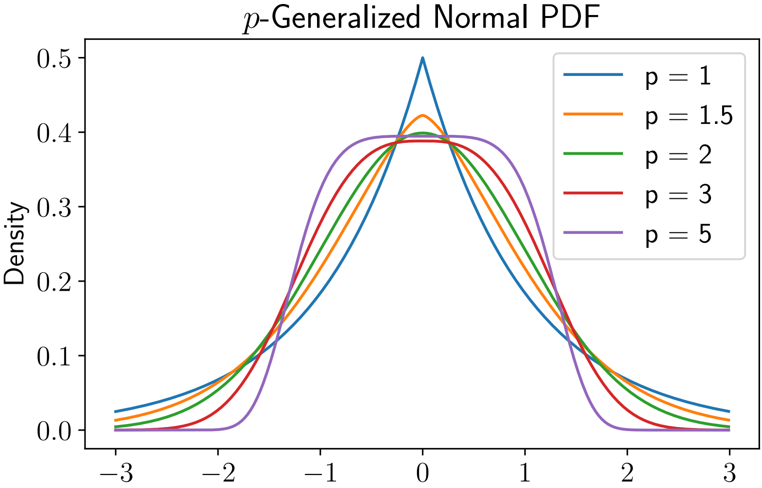
Use approximation ratio for evaluation:

- Let $\tilde{f}(\beta)$ be the loss function on the coreset and let $f(\beta)$ be the original loss
- Let β^{opt} be the solution of the original problem
- Optimize $\tilde{f}(\beta)$ to find solution $\tilde{\beta}$ of the reduced problem
- Compute ratio $\frac{f(\tilde{\beta})}{f(\beta^{opt})}$

Experiments

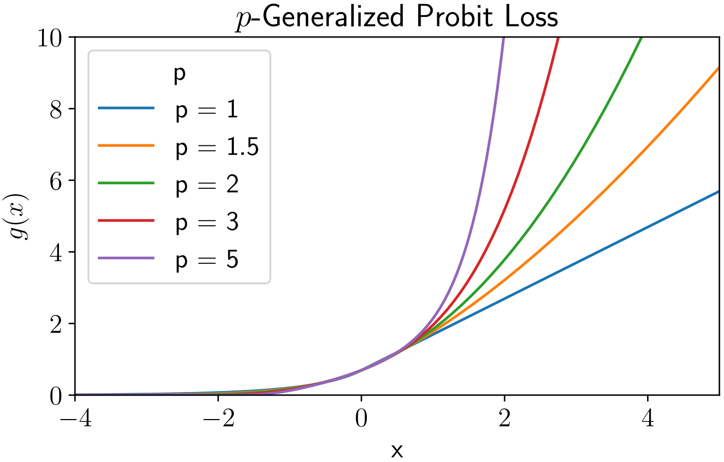


Extensions – p-Generalized Probit Model[3]



[3]See [Kalke and Richter, 2013] for a definition of the p-generalized normal distribution

Extensions – p-Generalized Probit Model



$$g(x) = \ln\left(\frac{1}{\Phi_p(-x)}\right)$$

Extensions – p-Generalized Probit Model

Three adaptations are required:

(1) Need to adapt μ:

$$\mu = \sup_{\beta \in \mathbb{R}^d \setminus \{0\}} \frac{\sum_{y_i x_i^T \beta > 0} |x_i^T \beta|^p}{\sum_{y_i x_i^T \beta < 0} |x_i^T \beta|^p}$$

Extensions – p-Generalized Probit Model

Three adaptations are required:

(1) Need to adapt μ:

$$\mu = \sup_{\beta \in \mathbb{R}^d \setminus \{0\}} \frac{\sum_{y_i x_i^T \beta > 0} |x_i^T \beta|^p}{\sum_{y_i x_i^T \beta < 0} |x_i^T \beta|^p}$$

(2) Need to use p-generalized leverage scores

Extensions – p-Generalized Probit Model

Three adaptations are required:

(1) Need to adapt μ:

$$\mu = \sup_{\beta \in \mathbb{R}^d \setminus \{0\}} \frac{\sum_{y_i x_i^T \beta > 0} |x_i^T \beta|^p}{\sum_{y_i x_i^T \beta < 0} |x_i^T \beta|^p}$$

(2) Need to use p-generalized leverage scores

(3) Need to adapt the sketching techniques

Extensions – p-Generalized Probit Model

Three adaptations are required:

(1) Need to adapt μ:

$$\mu = \sup_{\beta \in \mathbb{R}^d \setminus \{0\}} \frac{\sum_{y_i x_i^T \beta > 0} |x_i^T \beta|^p}{\sum_{y_i x_i^T \beta < 0} |x_i^T \beta|^p}$$

(2) Need to use p-generalized leverage scores

(3) Need to adapt the sketching techniques

Coreset size: $\begin{cases} O\left(\frac{\mu d^p}{\epsilon^2}\right), & for\ p \in [1,2) \\ O\left(\frac{\mu d^{2p}}{\epsilon^2}\right), & for\ p \in (2,\infty) \end{cases}$

Recap: Our Contributions

Recap: Our Contributions

(1) Enabled scalable maximum likelihood estimation of probit models on large datasets and data streams

- Based on fast coreset construction algorithms and modern sketching techniques

Recap: Our Contributions

(1) Enabled scalable maximum likelihood estimation of probit models on large datasets and data streams

- Based on fast coreset construction algorithms and modern sketching techniques

(2) Demonstrated that our methods outperform standard uniform sampling

- By conducting experiments on well known real-world datasets

Recap: Our Contributions

(1) Enabled scalable maximum likelihood estimation of probit models on large datasets and data streams

- Based on fast coreset construction algorithms and modern sketching techniques

(2) Demonstrated that our methods outperform standard uniform sampling

- By conducting experiments on well known real-world datasets

(3) Introduced the p-generalized probit model as a flexible framework for modeling binary data

- Enables you to control the tail behavior of the distribution

Literature I

Chao, M. T. (1982).
A general purpose unequal probability sampling plan.
Biometrika, 69(3):653–656.

Chhaya, R., Choudhari, J., Dasgupta, A., and Shit, S. (2020).
Streaming coresets for symmetric tensor factorization.
CoRR, abs/2006.01225.

Clarkson, K. L. and Woodruff, D. P. (2017).
Low-rank approximation and regression in input sparsity time.
J. ACM, 63(6).

Literature II

Feldman, D. and Langberg, M. (2011).
A unified framework for approximating and clustering data.
In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 569–578, New York, NY, USA. Association for Computing Machinery.

Feldman, D., Schmidt, M., and Sohler, C. (2020).
Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering.
SIAM Journal on Computing, 49(3):601–657.

Kalke, S. and Richter, W.-D. (2013).
Simulation of the p-generalized Gaussian distribution.
Journal of Statistical Computation and Simulation, 83(4):641–667.

Literature III

Langberg, M. and Schulman, L. J. (2010).
Universal ϵ-approximators for integrals.
In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 598–607.

Munteanu, A., Schwiegelshohn, C., Sohler, C., and Woodruff, D. P. (2018).
On coresets for logistic regression.
In *Advances in Neural Information Processing Systems 31, (NeurIPS)*, pages 6562–6571.