Project 3 - Statistical modelling and inference
Stochastic gradient descent

Felix Gutmann, Max Van Esso, Marco Fayet

December 24, 2015

1

To show: Fisher information matrix for the logit model

Prerequisite information:

- Definition fisher information: $\mathcal{I}(\theta) = -\mathbb{E}\left[\frac{\partial^2}{\partial\theta^2}\log f(X;\theta)\Big|\theta\right]$
- Input data for the model: $\boldsymbol{D} = \{(y_i, \boldsymbol{x}_i), \dots\}$
- Let y_i have binary outcome with distribution: $y_i \sim Bernoulli(p_i)$.
- The logit function of the model: $\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{x}_i^T\boldsymbol{\beta} \implies p_i = \frac{1}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}$

We will proceed as follows: First we will derive the Hessian Matrix of the logit model. After that we will apply the definition of the Fisher Information to obtain the result.

Solution:

We start from the Log-Likelihood function of the Bernoulli Distribution:

$$l = \sum_{i=1}^{N} y_i \log p_i + (1-y_i)\log(1-p_i)$$

We substitute p_i and rearrange it slightly

$$= \sum_{i=1}^{N} y_i \log\Big(\frac{1}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}\Big) + \log\Big(1 - \frac{1}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}\Big)(1-y_i)$$

$$= \sum_{i=1}^{N} y_i \log\Big(\frac{1}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}\Big) + \log\Big(\frac{e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}\Big)(1-y_i)$$

$$= \sum_{i=1}^{N} y_i \log\Big(\frac{1}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}\Big) + \Big(-\boldsymbol{x}_i^T\boldsymbol{\beta} - \log 1 + e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}\Big)(1-y_i)$$

Now we take the first derivative with respect to a certain parameter (β_j):

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{N} y_i x_{ij} \frac{e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}} + \Big(-x_{ij} + x_{ij}\frac{e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}{1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}\Big)(1-y_i)$$

Notice that the fraction expression is equal to $(1-p_i)$.

$$= \sum_{i=1}^{N} y_i x_{ij}(1-p_i) + \Big(-x_{ij} + x_{ij}(1-p_i)\Big)(1-y_i)$$

$$= \sum_{i=1}^{N} y_i x_{ij} - y_i x_{ij} p_i - x_{ij} p_i + y_i x_{ij} p_i$$

$$= \sum_{i=1}^{N} y_i x_{ij} - x_{ij} p_i$$

The last expression is an entry of the gradient. In order to find the Hessian we proceed by taking the second derivative

$$\frac{\partial^2 l}{\partial \beta_j \partial \beta_k} = -x_{ij} \frac{e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}}{(1+e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}})^2} x_{ij}$$

We simplify this expression to get an entry of the Hessian matrix

$$= -x_{ij} p_i (1-p_i) x_{ij}$$

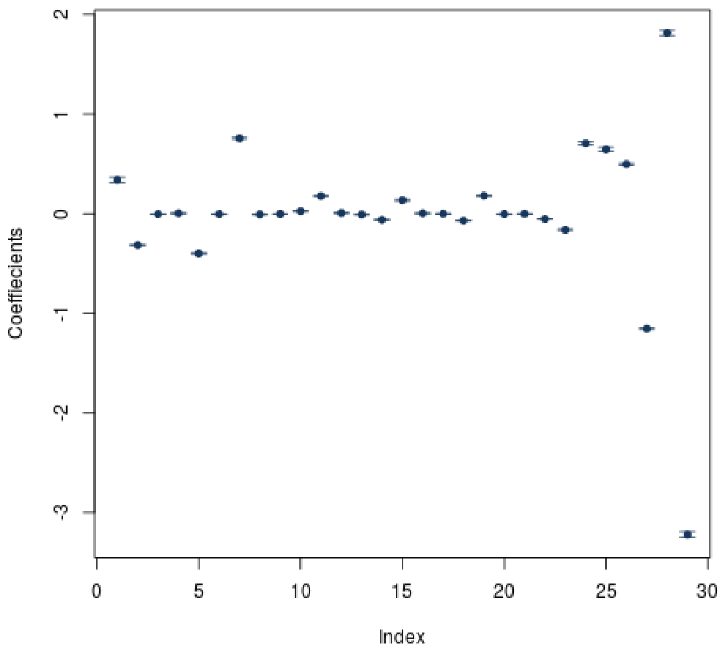
We express this in matrix terms to get the full Hessian Matrix:

$$\nabla_\beta^2 = -\boldsymbol{X}^T \boldsymbol{P} \boldsymbol{X}$$

, where $\boldsymbol{P}$ is diagonal matrix. Finally to get the Fisher Information we take the expected value of the Hessian with respect to y, which is not present and the expexted value vanishes:

$$\mathcal{I} = -\mathbb{E}\Big[-\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{X}\Big] = \boldsymbol{X}^T\boldsymbol{P}\boldsymbol{X}$$

2 Model coefficients with standard errors



3 Methodology and default implementation of the stochastic gradient descent in the sgd R package

The sgd R package estimates a vector of parameters $\theta_* \in \mathbb{R}^p$ using i.i.d. samples $\mathbf{D} = \{\mathbf{x}_n, \mathbf{y}_n\}$ with $n = 1, 2, ..., N$ and a log-likelihood function of the dataset D given by $\ell(\theta; \mathbf{D}) = \sum_{n=1}^{N} \log f(\mathbf{y}_n; \mathbf{x}_n, \theta)$.The averaged implicit stochastic gradient descent **(AI-SGD)** is the default implementation of the package. Similarly to other procedures, AI-SGD will have an exploration phase (where the iterate will rapidly approach θ^*) followed by a convergence phase (where the iterate will jitter around θ^*). With a running time complexity linear in the number of datapoints N but sublinear in the parameter dimension p, it is defined by the following procedure:

(1) $\theta_n^{im} = \theta_{n-1}^{im} + \gamma_n C_n \nabla \log f(\mathbf{y}_n; \mathbf{x}_n, \theta_n^{im})$

(2) $\bar{\theta}_n = (1/n) \sum_{i=1}^{n} \theta_i^{im}$

with $\gamma_n > 0$ the learning rate sequence and C_n a sequence of positive-definite matrices. The first step of this procedure illustrates the differences with the explicit **(E-SGD)**, with the implicit update θ_n^{im} appearing on both sides of the equation. The second step of this procedure, referred to as iterate averaging, garantees optimal statistical efficiency. Assuming a common starting point $\theta_{n-1}^{sgd} = \theta_{n-1}^{im} \triangleq \theta_0$ the implicit update is a shrinked version of the explicit update such that :

$$(3)\ \Delta\theta_n^{im} = (I + \gamma_n C_n \widehat{I}(\theta_0; \mathbf{x}_n, \mathbf{y}_n))^{-1} \Delta\theta_n^{sgd} + \mathcal{O}(\gamma_n^2)$$

with $\Delta\theta_n = \theta_n - \theta_{n-1}$, I the identity matrix, and $\widehat{I}(\theta_0; \mathbf{x}_n, \mathbf{y}_n)$ the observed Fisher information matrix at θ_0. This shrinkage effect stabilizes the iterations in small-to-moderate sized samples, and allows them to be more robust to misspecifications of the learning rate parameter γ. From a Bayesian standpoint, θ_n^{im} represents the posterior mode of a model with prior $\mathcal{N}(\theta_n^{im}, \gamma_n C_n)$ and likelihood $f(\theta; \mathbf{x}_n; \mathbf{y}_n)$.

The default implementation of the sgd R package applies to models whose likelihood $\ell(\theta; \mathbf{x}_n, \mathbf{y}_n) \equiv \log f(\mathbf{y}_n; \mathbf{x}_n, \theta)$ depends on θ only through the natural parameter $\eta \equiv \mathbf{x}^T\theta$, i.e. models for which $\ell(\theta; \mathbf{x}_n, \mathbf{y}_n) \equiv \ell(\mathbf{x}^T\theta; \mathbf{x}_n, \mathbf{y}_n)$. Under this assumption, the gradient for the implicit iterate θ_n^{im} is a scaled version of the gradient at the previous iterate. Computing the implicit update then reduces to finding this scale factor s_n. An elastic net regularization is performed on the implicit SGD via the addition of a penalty to the log-likelihood. With a regularization parameter $\lambda \in \mathbb{R}$, some fixed $\alpha \in [0,1]$ and a penalty function $P_\alpha(\theta) = (1-\alpha)\frac{1}{2}\|\theta\|_2^2 + \alpha\|\theta\|_1$, the implicit SGD update becomes:

$$(4)\ \theta_n^{im} = \theta_{n-1}^{im} + \gamma_n C_n (s_n \nabla \log f(\mathbf{y}_n; \mathbf{x}_n, \theta_{n-1}^{im}) - \lambda \nabla P_\alpha(\theta_{n-1}^{im}))$$

where the scale factor s_n satisfies :

$$s_n\kappa_{n-1} = \ell'(\mathbf{x}_n^T\theta_{n-1}^{im} - \gamma_n\lambda\mathbf{x}_n^T C_n \nabla P_\alpha(\theta_{n-1}^{im}) + \gamma_n s_n \kappa_{n-1} \mathbf{x}_n^T C_n \mathbf{x}_n; \mathbf{x}_n, \mathbf{y}_n)$$

with $\kappa_{n-1} = \ell'(\mathbf{x}_n^T.\theta_{n-1}^{im}; \mathbf{x}_n, \mathbf{y}_n)$

To summarize, the AI-SGD default procedure is computationally efficient, statistically optimal, and numerically stable. It run as follows:

Given the derivative of the model's loglikelihood $\ell'(\cdot;\cdot)$, a learning rate sequence γ_n, an estimation θ_{n-1}^{im}, sample data $\mathbf{x}_n, \mathbf{y}_n$, a sequence of positive-definite matrices C_n, and a penalty function P_α:

- First we establish the search bounds B such that $B = [0,\, r_n]$ if $r_n > 0$, and $B = [r_n,\, 0]$ otherwise, with $r_n = \gamma_n \ell'(\mathbf{x}_n^T\theta_{n-1}^{im}; \mathbf{x}_n, \mathbf{y}_n)$

- Then we use a root-finding method by solving

$$\Gamma = \ell'(\mathbf{x}_n^T\theta_{n-1}^{im} - \gamma_n\lambda\mathbf{x}_n^T C_n \nabla P_\alpha(\theta_{n-1}^{im}) + \Gamma\mathbf{x}_n^T C_n \mathbf{x}_n; \mathbf{x}_n, \mathbf{y}_n), \Gamma \in B$$

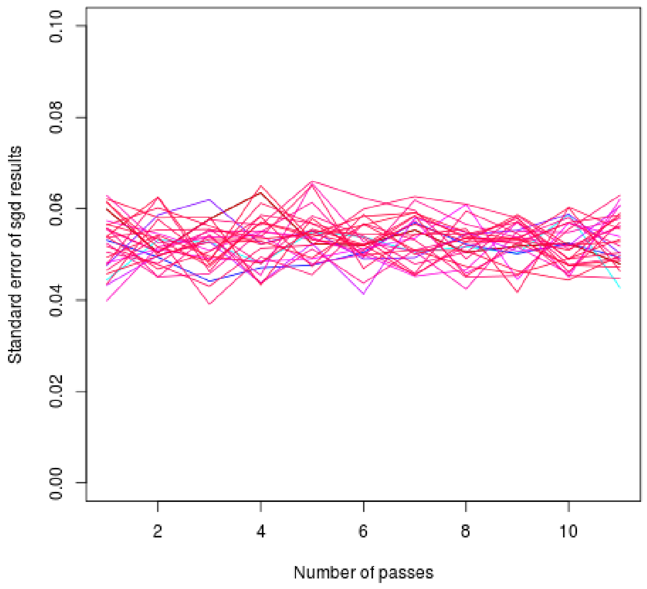
and assign $s_n = \Gamma/r_n$. This is equivalent to the implicit SGD as articulated in equation (4) and we can therefore get an updated iteration such that :

$$\theta_n^{im} = \theta_{n-1}^{im} + \gamma_n C_n (s_n \ell'(\mathbf{x}_n^T\theta_{n-1}^{im}; \mathbf{x}_n, \mathbf{y}_n)\mathbf{x}_n - \lambda\nabla P_\alpha(\theta_{n-1}^{im}))$$

4 SGD output

Remarks on results

Each line in the plot displays the standard error of an estimator as a function of the number of passes. We computed the standard errors of each estimator after 50 reruns for each number of passes. Initially we expected some kind of convergence, however our results don't show such a behaviour. Moreover we find bad approximations for the estimators. Untill now we haven't been able to find the underlying mistake. Hence, the only thing we can provide so far is the plot and a chunk of code displaying our approach (Listing 1: N - number of rows of the higgs data, cm01 - vector with MLE, c20 - empty storage matrix).



Listing 1: Code chunk - SGD

```
 for(i in 1:50){

 ...

 temp20  <- higgs[sample(N,N,replace=FALSE),]
 pass20  <- sgd(V1 ~ . , data = temp20 ,
             model = "glm",
             model.control=list(family=binomial(link="logit")),
             sgd.control=list( start=cm01, npasses =  20, pass = TRUE ) )
c20[i,] <- pass20$coefficients

...

}
```