PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2021, **55**(1), p. 29–35

M a t h e m a t i c s

LOSS FUNCTIONS AND DESCENT METHOD

V. K. OHANYAN *, H. Z. ZOHRABYAN **

Chair of the Theory of Probability and Mathematical Statistics, YSU, Armenia

In this paper, we showed that it is possible to use gradient descent method to get minimal error values of loss functions close to their Bayesian estimators. We calculated Bayesian estimators mathematically for different loss functions and tested them using gradient descent algorithm. This algorithm, working on Normal and Poisson distributions showed that it is possible to find minimal error values without having Bayesian estimators. Using Python, we tested the theory on loss functions with known Bayesian estimators as well as another loss functions, getting results proving the theory.

https://doi.org/10.46991/PYSU:A/2021.55.1.029

MSC2010: 91B30, 91G60, 62C10.

Keywords: Bayesian estimators, gradient descent, loss functions, machine learning.

Introduction. Nowadays Machine Learning is one of the most interesting fields of research in Computer Science. Being a sub-field of artificial intelligence, machine learning is generally used to learn on data inputs, identify certain patterns and make predictions or decisions by itself. A field, that's built on a foundation of mathematics and applied statistics performs tasks, that seemed impossible for a computer a few decades ago. Even though, machine learning is a field of computer science, it greatly differs from traditional programming approaches. In a usual programming approach, the code consists of rules and directions that computer follows consequently without ability to work otherwise or learn. Machine learning algorithms, allow computers to learn on the data inputs and make predictions based on them. Basic Machine learning tasks require optimizing the value of the prediction over one of the features in data that we want to predict. Optimization, on a feature can be interpreted as a problem of minimizing the value of the loss function defined on the predicted and actual values of feature. Unfortunately, there does not exist, a universally defined loss function which can be used with all models. Moreover, not all loss functions have defined minimum values or algorithms to find them. In this work, we will find minimal values of loss

* E-mail: victoohanyan@ysu.am

** E-mail: hovhannes.zohrabyan@yahoo.com, hovhannes_zohrabyan@edu.aua.am

function and show that descent method can be used to find minimal values of loss functions.

In machine learning there are three widely used basic loss functions. Those loss functions are the following: "Quadratic Loss", "Zero-One Loss" and "Absolute Loss" functions. We will discuss each of them separately, including their advantages and weaknesses in the next sections. One property that is common for all the above mentioned loss functions is that their minimal values can be computed using Bayesian estimators. We will show, how the minimum values for these functions can be found using Bayesian Estimation theory.

Bayesian inference derives the posterior density as a consequence of two antecedents: a prior density function $p(\theta)$ and a "likelihood function" $f(x_1,x_2,\ldots,x_n|\theta)$ derived from a statistical model for the observed data. Using Bayes theorem we can define the relationship between posterior density and likelihood functions (see [1]):

$$p(\theta|x_1,x_2,\ldots,x_n) = \frac{f(x_1,x_2,\ldots,x_n|\theta)}{p(x_1,x_2,\ldots,x_n)}p(\theta), \tag{1}$$

where $p(\theta|x_1,x_2,\ldots,x_n)$ is posterior density function and $p(x_1,x_2,\ldots,x_n)$ is the marginal density of data.

A random experiment is used to generate measurements $x_1,x_2,...,x_n$ from $f(x_1,x_2,\ldots,x_n|\theta)$, where the parameter θ has a density function $p(\theta)$. Our aim is to estimate θ from $x_1,x_2,...,x_n$. After, we denote the estimator by $\hat{\theta} = \hat{\theta}(x_1,x_2,...,x_n)$. By definition any measurable function of sample $x_1,x_2,\ldots,x_n$ is called an estimator.

Loss Function and Risk. Our aim is to estimate $\hat{\theta}$, which has the minimal loss from the actual value θ. The difference or loss, between our estimator and unknown parameter θ is presented with a loss function $L(\theta,\hat{\theta})$.

However, in this form this function is not suitable, as both arguments are random variables. In order to work with difference of random variables, we can take the expected value and try to minimize it. One of the ways that we can take to define the expected value function is to average over θ. In the other case we will define the value of expected loss function over the $x_1,x_2,\ldots,x_n$ (see [2] and [3]).

$$E(L(\theta,\hat{\theta})|x_1,x_2,\ldots,x_n) = \int L(\theta,\hat{\theta})p(\theta|x_1,x_2,\ldots,x_n)d\theta. \tag{2}$$

On the other hand we can average given θ over $x_1,x_2,\ldots,x_n$. In this case we get an expected function defined in the following way

$$E(L(\theta,\hat{\theta})|\theta) = \int L(\theta,\hat{\theta})f(x_1,x_2,\ldots,x_n|\theta)dx_1\ldots dx_n.$$

In the Bayesian problem setup we will use (2) and call it the Bayesian Expected Loss function. But a more important function in the Bayesian setup is the risk function. Generally, the risk is defined as an average loss function over the $f(x_1,x_2,...,x_n|\theta)$ (likelihood function) (see [3] or [4]), defined as follows

$$\begin{aligned} R(\theta,\hat{\theta}) &= \int E(L(\theta,\hat{\theta})|\theta)p(\theta)d\theta \\ &= \int \left[\int L(\theta,\hat{\theta})f(x_1,x_2,\ldots,x_n|\theta)dx_1\ldots dx_n\right] p(\theta)d\theta. \end{aligned}$$

$R(\theta,\hat{\theta})$ is the general function for the risk, for the Bayesian setup the notion of risk is changed. Bayesian risk is the risk averaged on the prior density. Recall the definition of Bayesian risk, given by

$$R(\theta,\hat{\theta}) = \int\int L(\theta,\hat{\theta})p(x_1,x_2,\ldots,x_n,\theta)dx_1\ldots dx_n d\theta, \quad \text{where}$$
$$p(x_1,x_2,\ldots,x_n,\theta) = f(x_1,x_2,\ldots,x_n|\theta)p(\theta).$$

Now we can apply (1) and get

$$p(\theta|x_1,x_2,\ldots,x_n)p(x_1,x_2,\ldots,x_n) = p(\theta)f(x_1,x_2,\ldots,x_n|\theta).$$

When Bayes rule is applied in the formula above, we get a function of posterior density of θ given by $x_1,x_2,\ldots,x_n$ and joint density function $p(x_1,x_2,\ldots,x_n,\theta)$ of data $x_1,x_2,\ldots,x_n$ and unknown parameter θ.

This formula has a very important interpretation. In this context we say that the prior density is mapped to the posterior density (see (1))

$$p(\theta) \to p(\theta|x_1,x_2,\ldots,x_n).$$

Thus we can interpret the Bayes Risk estimator in the following way

$$\hat{\theta}(x_1,x_2,\ldots,x_n) = \underset{\hat{\theta}}{\operatorname{argmin}} \int L(\theta,\hat{\theta})p(\theta|x_1,x_2,\ldots,x_n)d\theta, \tag{3}$$

where argmin are the points of the function domain at which the values are minimized.

Loss Functions.

Quadratic Loss Function. Quadratic loss function, is defined as follows

$$L(\theta,\hat{\theta}) = [\theta-\hat{\theta}]^2. \tag{4}$$

Quadratic Loss function is one of the most common loss functions used with regression problems in Machine Learning. However, this function comes with a serious disadvantage. Quadratic Loss function is very sensitive to outliers (because we use power 2). As a result, if there are outliers in the training dataset, we can get great errors. , In order to minimize the value of quadratic loss function, we substitute (4) in (3). We get

$$\int L(\theta,\hat{\theta})p(\theta|x_1,x_2,\ldots,x_n)d\theta = \int [\theta-\hat{\theta}]^2 p(\theta|x_1,x_2,\ldots,x_n)d\theta.$$

After getting the formula for Bayes risk in case of Quadratic Loss function, we can find it's first derivative. That is

$$\frac{\partial}{\partial\hat{\theta}}\int [\theta-\hat{\theta}]^2 p(\theta|x_1,x_2,\ldots,x_n)d\theta = -2\int [\theta-\hat{\theta}]p(\theta|x_1,x_2\ldots,x_n)d\theta.$$

And using that, we calculate

$$-2\int [\theta-\hat{\theta}]p(\theta|x_1,x_2,\ldots,x_n)d\theta = 0, \quad \text{and so}$$
$$\hat{\theta}(x_1,x_2,\ldots,x_n) = \int \theta p(\theta|x_1,x_2,\ldots,x_n)d\theta.$$

Since the second derivative is equal 2, we come to the following result.

Theorem 1. [2]. *For the Quadratic loss function the Bayes estimator is equal to the mean of the posterior density function.*

Zero-One Loss Function. Zero-One loss function is defined for $\varepsilon > 0$ as follows

$$L(\theta,\hat{\theta}) = \begin{cases} 0, & \text{if } |\theta-\hat{\theta}| \le \varepsilon, \\ 1, & \text{if } |\theta-\hat{\theta}| > \varepsilon. \end{cases}$$

We can calculate

$$\begin{aligned} E(L(\theta,\hat{\theta})) &= P(|\theta-\hat{\theta}| > \varepsilon) \\ &= 1 - P(|\theta-\hat{\theta}| \le \varepsilon) = 1 - \int\limits_{\hat{\theta}-\varepsilon}^{\hat{\theta}+\varepsilon} p(\theta|x_1,x_2,...,x_n)d\theta. \end{aligned} \tag{5}$$

We can see that the value of the above computed Expectation function can be minimized if the value of integral is maximized

$$\hat{\theta}(x_1,x_2,\ldots,x_n) = \operatorname*{argmax}_{\theta} \int\limits_{\hat{\theta}-\varepsilon}^{\hat{\theta}+\varepsilon} p(\theta|x_1,x_2,\ldots,x_n)d\theta.$$

Theorem 2. *For the Zero-One loss function the Bayes estimator is equal to the mode of the posterior density function.*

Absolute Error Loss Function. Absolute error loss function is defined as

$$L(\theta,\hat{\theta}) = |\hat{\theta}-\theta|.$$

Using the definition of expected value, we get

$$\begin{aligned} E(|\hat{\theta}-\theta|) &= \int |\hat{\theta}-\theta| p(\theta|x_1,x_2,\ldots,x_n)d\theta \\ &= \int\limits_{-\infty}^{\hat{\theta}} (\hat{\theta}-\theta) p(\theta|x_1,x_2,\ldots,x_n)d\theta + \int\limits_{-\infty}^{\hat{\theta}} (\theta-\hat{\theta}) p(\theta|x_1,x_2,\ldots,x_n)d\theta. \end{aligned}$$

By calculating the first derivative in $\hat{\theta}$ and equating to 0, we will get

$$\int\limits_{-\infty}^{\hat{\theta}} p(\theta|x_1,x_2,\ldots,x_n)d\theta = \int\limits_{\hat{\theta}}^{\infty} p(\theta|x_1,x_2,\ldots,x_n)d\theta,$$

and after this we can see that

$$2\int\limits_{-\infty}^{\hat{\theta}} p(\theta|x_1,x_2,\ldots,x_n) = \int\limits_{-\infty}^{\infty} p(\theta|x_1,x_2,\ldots,x_n) = 1.$$

That is

$$\int\limits_{-\infty}^{\hat{\theta}} p(\theta|x_1,x_2,\ldots,x_n) = \frac{1}{2}.$$

The last equation implies that $\hat{\theta}$ is a median of the posterior density.

Theorem 3. *For the Absolute loss function the Bayes estimator is equal to a median of the posterior density function.*

Simulating Problem using Python. In order to test the theory and make calculations for new cases we need to set up our problem and environment. For the simulation purposes we try to estimate mean for Normal and Poisson distributions using different loss functions. Beforehand, the theory has to be checked using a loss function with known Bayes estimator. Given that the results are good, other loss functions can be used. Python will be used as a programming language, alongside "Numpy" and "Scipy" packages used to generate data and calculating derivatives of the loss functions. To simulate the theory, for each estimated $\hat{\theta}$ the random data $y_1, y_2, \ldots, y_n$, of length n from Normal or Poisson Distribution will be generated.

Simulating for Quadratic Loss Function. In order to work with data, loss function will be defined as follows

$$L(\theta, \hat{\theta}) = \sum_{k=1}^{n} (x_k - y_k)^2,$$

where $y_1, y_2, \ldots, y_n$ is data generated using estimated $\hat{\theta}$.

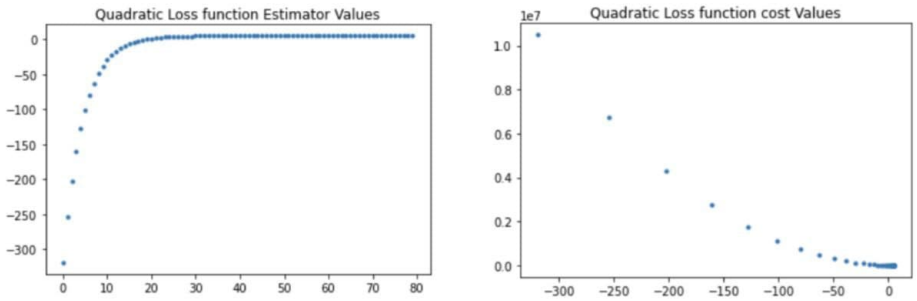

Fig.1. Quadratic Loss function report, where the left graphic illustrates the values of estimators before converging and the right graphic illustrates the decrease of loss for the estimator.

For the Quadratic loss function Bayes estimator is equal to the mean of the posterior density function (see Theorem 1). We will use "Scipy" package in order to compute the posterior density function. For the data $x_1, x_2, \ldots, x_n$ the mean of the posterior density function is equal to 0.0499. Which means, that the gradient descend algorithm should get minimal error value around 0.0499. In order to check that, we will construct a gradient descend algorithm with the Quadratic Loss function.

In order to find the estimator minimizing loss function, derivative at each point will be taken [4]. Derivative will give the direction to either increase or decrease estimator $\hat{\theta}$. Mathematically the gradient descent will have the following form

$$\hat{\theta}_i := \hat{\theta}_{i-1} - \alpha \frac{\partial L(\hat{\theta})}{\partial \theta},$$

where α is the learning rate, x_k is the data sample, and y_n is data generated using the estimator.

These algorithm is repeated until difference between $\hat{\theta}_i$ and $\hat{\theta}_{i-1}$ is so small, that the values are basically the same (see [5]). Finally, the algorithm converges with the minimum error 0.01. This minimal error is close to the Bayesian estimate.

Using the Quadratic loss function, the algorithm will be tested on data from

Poisson distribution as well. In the case of Poisson distribution the minimal error is equal to 0.02, which is close to the Bayesian estimate as well.

Absolute Loss Function. We will run the same simulation for the absolute loss function, which is defined as follows

$$L(\theta,\hat{\theta}) = \sum_{k=1}^{n} |x_k - y_k|.$$

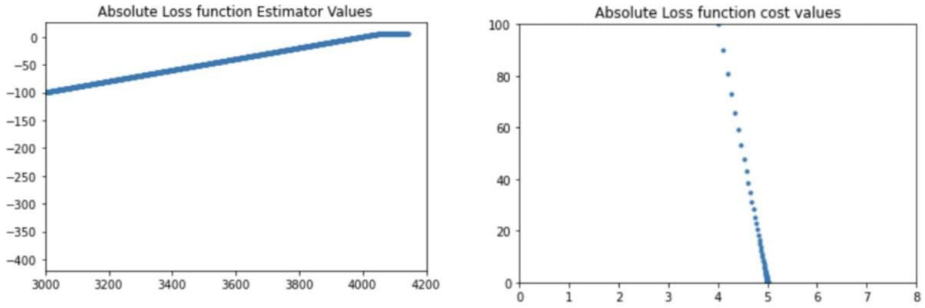

Fig. 2. Absolute Loss function report, where the left graphic illustrates the values of estimators before converging and the right graphic illustrates the decrease of the loss for the estimator.

For the absolute loss function the Bayesian estimate is the median of the distribution function, which in our case is equal to 10^{-8} for the Normal Distribution. Using this loss function the error in the gradient descent is equal to $175 \cdot 10^{-8}$. For the Poisson distribution the minimal error is equal 0.18. As a result, we show that it is possible to get comparable loss values to Bayesian estimates, using gradient descend.

Log-Cosh Loss Function. As we have shown, the gradient descent can be estimated, approaching the actual Bayesian estimates. As a result, we can use the gradient descend on the loss function, which does not have a defined Bayesian estimate. For this case we will choose the loss function of logarithm of the hyperbolic cosine called Log-Cosh and defined as follows

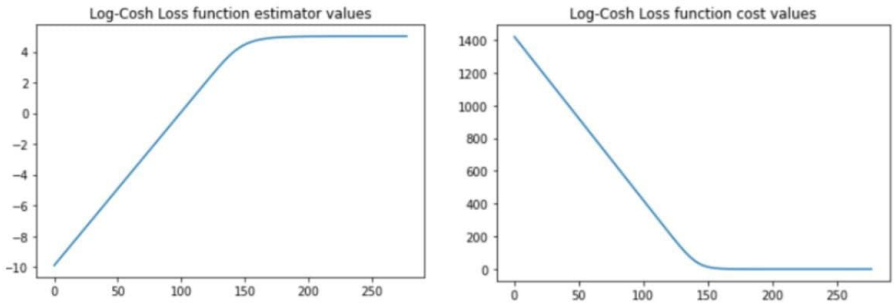

Fig. 3. Log-Cosh Loss function report, where the left graphic illustrates the values of the estimators before converging and the right graphic illustrates the decrease of loss for the estimator.

$$L(\theta,\hat{\theta}) = \sum_{k=1}^{n} \log(\cosh(x_k - y_k)). \tag{6}$$

For this loss function we run the gradient descent and get a minimum error equal to $232 \cdot 10^{-7}$.

Received 07.04.2021
Reviewed 23.04.2021
Accepted 27.04.2021

REFERENCES

1. Gelman A., Carlin J.B., Stern H.S., Dunson D.B., Vehtari A., Rubin D.B. *Bayesian Data Analysis. Chapman and Hall/CRC Texts in Statistical Science.* (3rd ed.). (2013).
2. Hasan M.R., Baizid A.R. Bayesian Estimation under Different Loss Functions Using Gamma Prior for the Case of Exponential Distribution. *Journal of Scientific Research. Section A: Physical and Mathematical Sciences* **9** : 1 (2017). https://doi.org/10.3329/jsr.v1i1.29308
3. Donovan Th.M., Mickey R.M. *Bayesian Statistics for Beginners: a Step-by-step Approach*. (2018). https://doi.org/10.3389/fpsyg.2020.01017
4. Hastie T., Tibshirani R. *The Elements of Statistical Learning.* (2008). https://doi.org/10.1007/978-0-387-84858-7
5. Martin O. *Bayesian Analysis with Python: Introduction to Statistical Modeling and Probabilistic Programming using PyMC3 and ArviZ.* (2018).

Վ. Կ. ՕՀԱՆՅԱՆ, Ռ. Զ. ԶՈՀՐԱԲՅԱՆ

ԿՈՐՍՏԻ ՖՈՒՆԿՑԻԱՆԵՐ ԵՎ ԱՆԿՄԱՆ ՄԵԹՈԴ

Հոդվածում ցույց ենք տվել, որ հնարավոր է օգտագործել աստիճանական իջման ալգորիթմ կորստի ֆունկցիաների նվազագույն սխալների արժեքներ ստանալու համար, որոնք մոտ են իրենց բայեսյան գնահատականներին: Տվյալ ալգորիթմի գործարկումը, որը աշխատում է "Նորմալ" և "Poisson" բաշխումների վրա, ցույց տվեց, որ հնարավոր է գտնել նվազագույն սխալների արժեքներ՝ չունենալով բայեսյան գնահատականներ: Կիրառելով Python-ը՝ փորձարկել ենք հայտնի բայեսյան գնահատականներով կորստի ֆունկցիաները, ինչպես նաև այլ կորստի ֆունկցիաներ, ինչի արդյունքում ստացել ենք տեսությունն ապացուցող արդյունքներ:

В. К. ОГАНЯН, О. З. ЗОГРАБЯН

ФУНКЦИИ ПОТЕРЬ И МЕТОД СПУСКА

В статье мы показали, что можно использовать алгоритм градиентного спуска для получения минимальных значений ошибок функций потерь, близких к их байесовским оценкам. Этот алгоритм, работающий на основе нормального и пуассоновского распределений, показал, что можно найти минимальные значения ошибок без байесовских оценок. Используя Python, мы протестировали теорию функций потерь с известными байесовскими оценками, а также другие функции потерь и получили результаты, подтверждающие теорию.